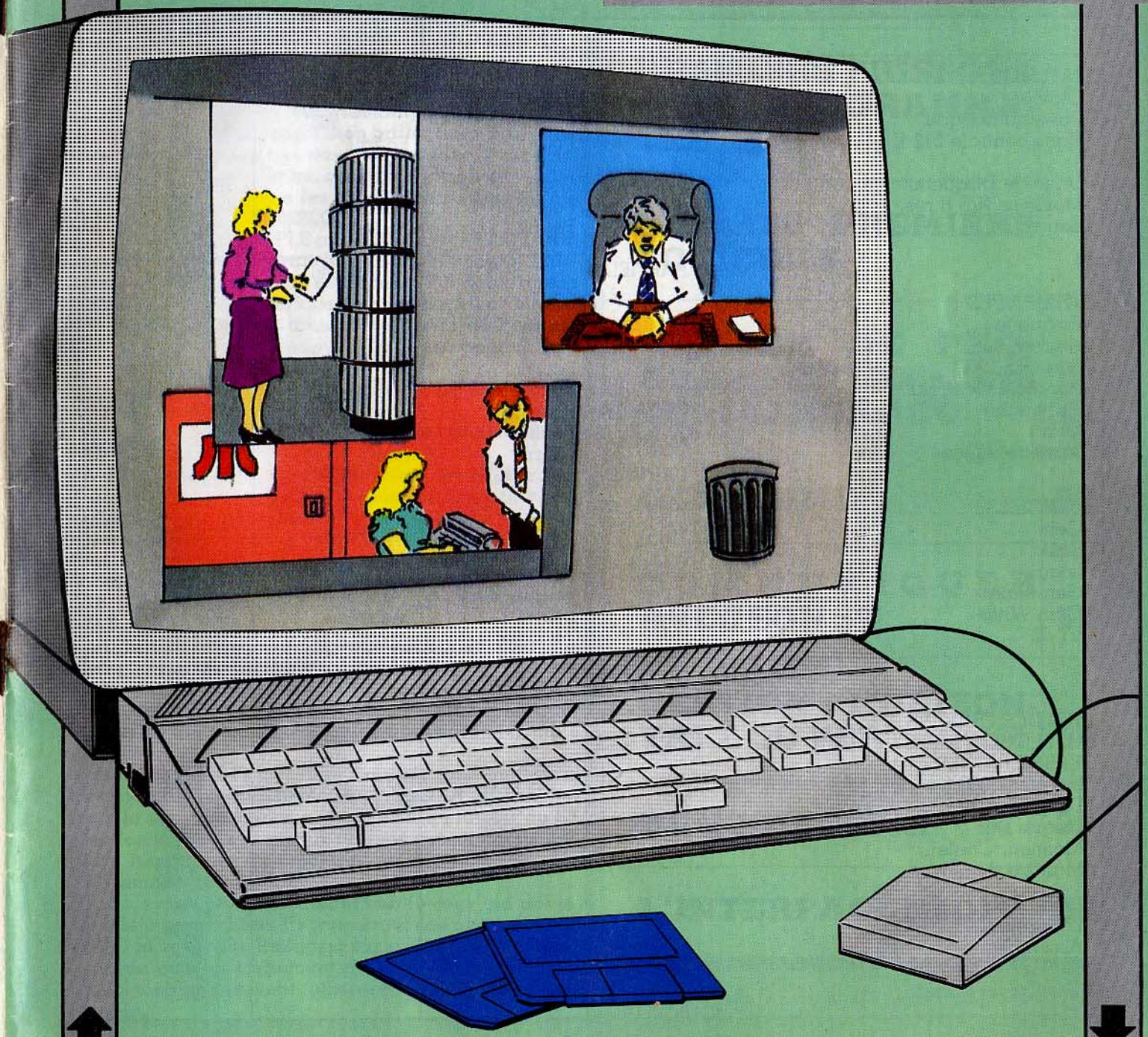
N° 1

SEPTEMBRE 1985

ST magazine

POURQUOI ACHETER UN 520 ST



25 F - Belgique 200 FB - Suisse 10 FS

UNE REVOLUTION DANS LE PRIX

Pour vous aider à mieux percevoir la nouvelle étape franchie avec le ST vers une informatique abordable, nous calculons dans ce tableau ce que vous devez dépenser pour un MacIntosh ou un IBM PC épuipés des mêmes options et des mêmes logiciels que ceux qui sont inclus dans la configuration 520 ST à 8432 Francs H.T.
Les prix pour le McInstosh et l'IBM PC sont approximatifs, ils dépendent des boutiques. Pour chaque logiciel ou périphérique, s'il existe deux ou plusieurs alternatives, nous avons choisi la moins chère.
Nous avons essayé de rendre ce tableau le plus exact possible, nous vous demandons à l'avance de nous excuser pour toute erreur ou omission qui pourrait s'y trouver. Mais les différences de prix sont telles qu'une erreur même de 1000 ou 2000 francs ne change rigoureusement rien à l'affaire.
Dernière minute : Au moment de mettre sous presse de fortes rumeurs de baisse de prix sur le McIntosh courent dans les milieux spécialisés. Merci Atari !

	ATARI 520 ST	McINSTOSH	IBM PC
MATERIEL :			
Clavier	Fourni	Fourni	2585
Pavé numérique	Fourni	965	Fourni
Unité centrale 512 K	Fourni	Fourni	16005
			8320
Unité de Disquette	(360 K)	(400 K)	(320 K)
Moniteur N/B h.res.	Fourni	Fourni	2590
Souris	Fourni	Fourni	2400
TOTAL PARTIEL :	8432	30940	31800
INTERFACES :			
Disquette	Fourni	Fourni	Fourni
Disquet dur	Fourni	non	
Série RS 232	Fourni	2	1205
Parallèle Centronics	Fourni	3000	2515
Cartouche	Fourni	non	non
Midi	Fourni	non	non
Manette de jeu	Fourni	non	445
TOTAL PARTIEL :	8432	33960	35975
LOGICIELS :			
Gem	Fourni	Equivalent	1000
Basic	Fourni	1800	1500
Logo	Fourni	1500	2050
Gem Paint	Fourni	MacPaint	non
Gem Write	Fourni	MacWrite	non
VT 52			
TOTAL DEFINITIF	8432	37260	40525
PRIX DES OPTIONS SUPPLEMENTAIRES :			
PERIPHERIQUES :			
Disquette 360 K	1800	4635	4175
Disquette 720 K	2400	Non	Non
Disque Dur	6000	24000	14000
Moniteur Couleur	4500	Non	6830
Manette de jeu	100	Non	675
LOGICIELS :			
Langage C	700	5000	5000
Assembleur	700	1345	1850
Tableur	1300	2210	2775
Gestion de fichier	1300	3325	2810

Comité de rédaction : Philippe GIUDICELLI. Jean-Michel DUBOIS. Godefroy GIUDICELLI.
Directeur de la publication : Godefroy GIUDICELLI. Rédacteur en chef : Philippe GIUDICELLI. Directeur technique : Jean-Michel DUBOIS.
Ont collaboré à ce numéro : J. NAVIA - J.F. MOUSSEAU - J.F. RAPIN - J. JACOB
ST MAGAZINE est édité par PRESSIMAGE, S.A.R.L. de presse au capital de 2.000 francs. 210, rue du Faubourg St-Martin, 75010 PARIS
Dépôt légal : 3e Trimestre 85. Tarif de l'abonnement : 200 francs (10 numéros).

EDITORIAL

Pourquoi acheter le 520ST ?
Nous avons voulu apporter la réponse dans ces pages à ceux de nos lecteurs qui n'ont pas encore la machine.
Si nous lançons aussitôt un magazine entièrement consacré au ST, c'est que nous pensons que cette machine est le moteur d'une révolution au moins aussi importante que le fût celle de l'APPLE II en son temps.
PAR LE PRIX :
Une configuration de base 512K Ram est déjà proposée à un prix que beaucoup jugeaient impossible lorsqu'il a été annoncé il y a quelques mois. Mais si on s'y penche de plus près, c'est encore plus extraordinaire. Voyez notre tableau comparatif avec McIntosh et IBM PC.
PAR LA PUISSANCE :520K Ram de base, ce sera le standard dans quelques années. Cette mémoire est maintenant disponible au plus grand nombre. Qu'en résultera-t-il au niveau de la qualité des programmes ? Réponse dans quelques mois !
PAR LA POLYVALENCE :
Sa puissance, son prix, ses possibilités graphiques en font un outil qui peut aussi bien briller dans le logiciel bureautique integré que dans le jeu vidéo. Pour vous faire une idée plus précise, consultez notre liste des 150 logiciels attendus dans les trois prochains mois.
PAR LA FACILITE D'EMPLOI :
XEROX a defriché, APPLE a fait connaître, ATARI rend abordable. Telle pourrait être la fable contant l'histoire de l'ordinateur et de ses icônes. Nul doute que la simplification de l'interface homme/machine soit une des conditions de la diffusion de masse de l'ordinateur. Vous en saurez plus sur GEM et ses applications dans le présent numéro.
PAR LES NOUVEAUX CHALLENGES OFFERTS AUX PROGRAMMEURS :
Sur les machines 64K, l'alternative est simple. Basic ou/et Assembleur.
Les autres langages sont marginaux, car le plus souvent limités par la mémoire disponible. Les prochains mois vont voir l'émergence du langage C, voire du Pascal libérant les programmeurs des tracas du passage à l'assembleur. Vous trouverez un premier article et des listings en C dans notre cahier central.
PAR DES NOUVEAUX UTILISATEURS PROFESSIONNELS :
L'utilité d'un ordinateur dans une entreprise n'est plus niée que par une poignée d'irréductibles. Ce qui empêche une généralisation de cet outil, c'est son coût. Jadis le privilège de très grosses entreprises, les ordinateurs se trouvent aujourd'hui dans un grand nombre de petites.
A moins de 15000 Francs Hors Taxe pour une configuration avec disque dur 10 Mega, il descend aujourd'hui dans la rue. Nous vous parlons ici de l'intérêt du ST pour les logiciels verticaux et nous consacrerons une place importante du prochain numéro aux premiers programmes disponibles.
Nous n'avons pas pour autant oublié ceux qui possèdent déjà la machine. Ils trouveront des articles sur le LOGO et le C, des listings à taper, les bancs d'essai d'un jeu et d'un traitement de texte. Et comment se connecter à un des nombreux serveurs maintenant disponible.
La fête commence !

SOMMAIRE

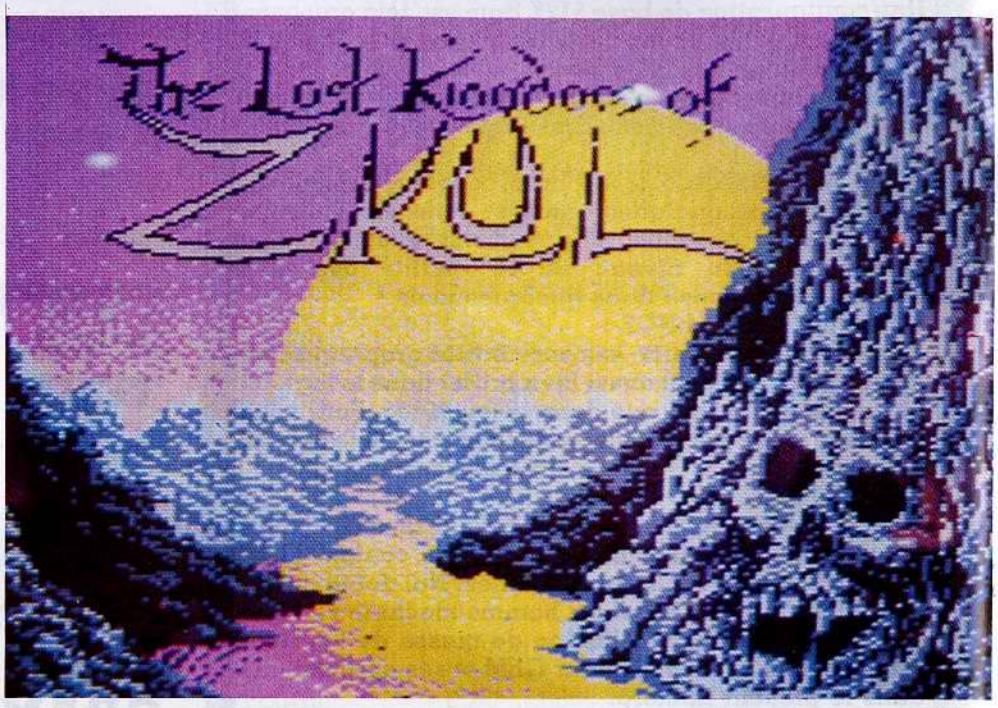
ST : LE PRIX D'UNE REVOLUTION	2
REPORTAGE DU PCW SHOW : LA FETE AU ST	4
PREMIERS FRISSONS D'UN PROGRAMMATEUR	6
LES 150 PREMIERS LOGICIELS	8
CAHIER DE PROGRAMMATION : LOGO et LANGAGE C	13
GEM J'AIME	26
COMMUNIQUER AVEC SON ST	30
GESTION : HORIZONTALE OU VERTICALE	34
BANCS D'ESSAI	36
L'INTERFACE MIDI	38

Le laps de temps dont nous avons disposé pour préparer ce numéro vous fera comprendre que nous ayions du privilégier le fond par rapport à la forme.

LA FÊTE AU ST

Le PCM Show qui ouvre traditionnellement la saison informatique s'est tenu à Londres la première semaine de septembre. C'était l'occasion révée de faire le point sur l'accueil reservé au 520 ST au moment même ou il devient disponible en Europe. D'autant que les premieres rumeurs indiquaient un support important de la part des sociétés de soft anglaises.

Et bien du soft, on en a vu !
Le stand Atari était le plus important de tout le Salon, mais Atari lui même n'en occupait qu'un quart environ. Le reste était occupé par des batteries de ST sur lesquels de nombreuses sociétés présentaient leurs premiers programmes pour le 520.
Très peu de nouveautés côté matériel. Un seul 260 ST dans un cube de verre, pas de CD Rom, aucun nouveau matériel. Les nouveautés, si il y en a, seront présentées au COMDEX de Las Vegas en novembre. Il s'agirait d'un matériel beaucoup plus puissant, un ordinateur 32 bits type VAX, ATARI désirant mettre la puissance d'un mini dans un micro qui coûterait moins de 5000 dollars.
En attendant on se consacre à ce qui existe, c'est-à-dire la ligne 520 ST. Pour ce faire, ATARI UK n'avait pas ménagé ses efforts, permettant à une vingtaine de sociétés de soft de venir s'installer sur son stand. Dans une approche qui confirme l'éclectisme des sociétés intéressées, s'y cotoyait de grosses sociétés de soft professionnel et de petites sociétés fraîchement créées ou l'amateurisme le disputait à l'enthousiasme.
Un distributeur anglais, spécialisé sur ATARI, avait même recréé une mini-boutique sur le stand, où il présentait les dernières nouveautés pour la gamme 8 bits et les premiers programmes disponibles pour les ST. Cette vitrine était en évolution pendant le Show même, puisque nous avons pu avoir un arrivage des Etats-Unis le soir de notre départ avec des titres d'Infocom et de Mirage.



Beaucoup de langages de programmation et de logiciels professionnels, du graphisme et des jeux, tel était le menu offert aux journalistes et aux revendeurs seuls admis lors de ces journées professionnelles.

LA PROGRAMMATION

Les sociétés anglaises ont beaucoup investi dans le QL de Sinclair, ce qui leur donnent un avantage important pour des produits à destination du 68000. Ce n'est donc pas une surprise de retrouver des sociétés connues pour leur support du QL derrière un grand nombre des outils de développement montré à Londres. C'est le cas de GST qui adapte pour le ST son C Compiler et son Macro-Assembleur 68000. Ces outils sont modifiés pour pouvoir utiliser l'environnement GEM.
Une autre leçon de ce salon réside dans l'émergence des compilateurs C comme langage dominant pour le dévellopement sur ST. Trois versions de ce langage (GST, MATACOMCO, COMPUTEUR ONE) étaient annoncées qui viennent s'ajouter au 4 ou 5 devant arriver des Etats-Unis.
Des Assembleurs, du Lisp, du Pascal et même du Fortran viennent compléter ce panorama ainsi que BBAS, un interpreteur Basic qui pourrait lire la plupart des programmes écrits pour le BBC avec des fonctions pour gérer les fenêtres et l'interface MIDI.
Nouveautés enfin sur le front des systèmes d'exploitation. Certaines sociétés anglaises délaissant le CP/M 68K (TOS), faisait tourner le ST sous BOS (Business Operating System) et sous P-System. Ce qui permet au ST de disposer dès maintenant d'un grand nombre de programmes, notamment professionnels. GST aurait également en préparation un émulateur CP/M.

LE PROFESSIONNEL

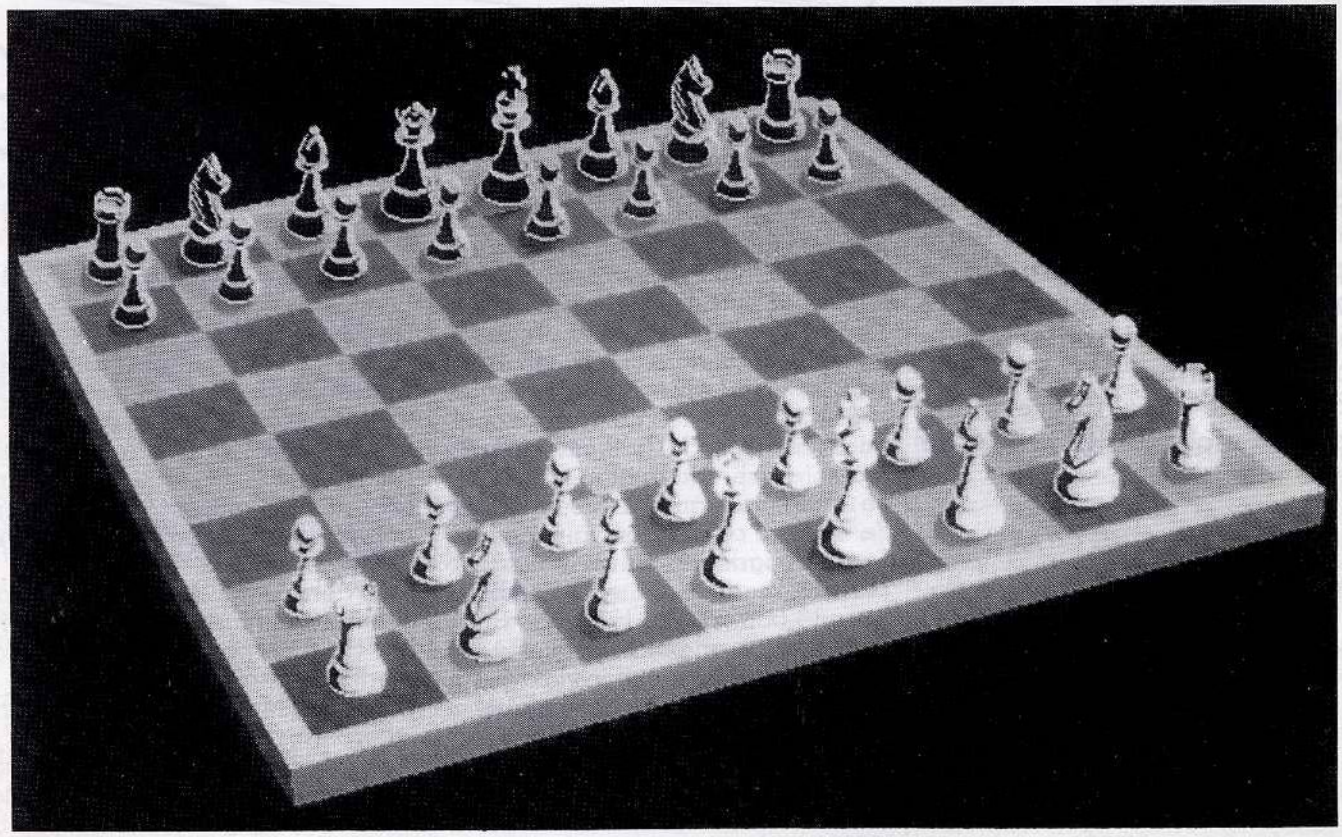
Pas d'intégrés chez nos amis anglais, mais généralisation du tableur a 49.95$ (CHEZ KUMA et SOFTWARE PUNCH). Même prix pour le gestionnaire de fichiers ou le traitement de texte. Ces logiciels utilisent GEM, même si pour beaucoup ils sont transcrits à partir d'autres machines. Programmes de PSION qui viennent du QL. de SINCLAIR, programmes de SOFTECHNIES (Timelink, Rhythm) du Mc Intosh.
Enfin, le réseau ATARINET de SOFTWARE PUNCH permet à 250 utilisateurs de communiquer en réseau local.
On peut dire que dès ce mois-ci, les utilisateurs anglais ont l'embarras du choix quand à la sélection de logiciels professionnels. En effet à la somme de tableurs, traitements de texte, gestionnaires de fichiers et logiciels de comptabilité divers s'ajoutent les dizaines de programmes disponibles sous BOS ou sous P-System. Malheureusement une grande partie de ces logiciels ne sera pas utilisable chez nous.

LE GRAPHISME

Peu de logiciels graphiques pour le moment, mis à part GEM Draw abondamment démontré sur de nombreux ST par des gens de chez DIGITAL.
C'est bien normal, pour exploiter et GEM, et le graphisme nouveau du ST, il faut plus de temps que pour adapter des langages venant d'autres machines. Par contre les annonces sont nombreuses, parmi lesquelles nous avons noté un logiciel de photocomposition « Fleet Street Editor » de chez Mirrorsoft.
A noter pour finir une démo époustouflante d'images digitalisées qui défilaient à toute allure sur l'écran d'un ST. L'accès en était interdit ; aussi aucune information sur la manière dont elles ont été générées n'a pu être connue.

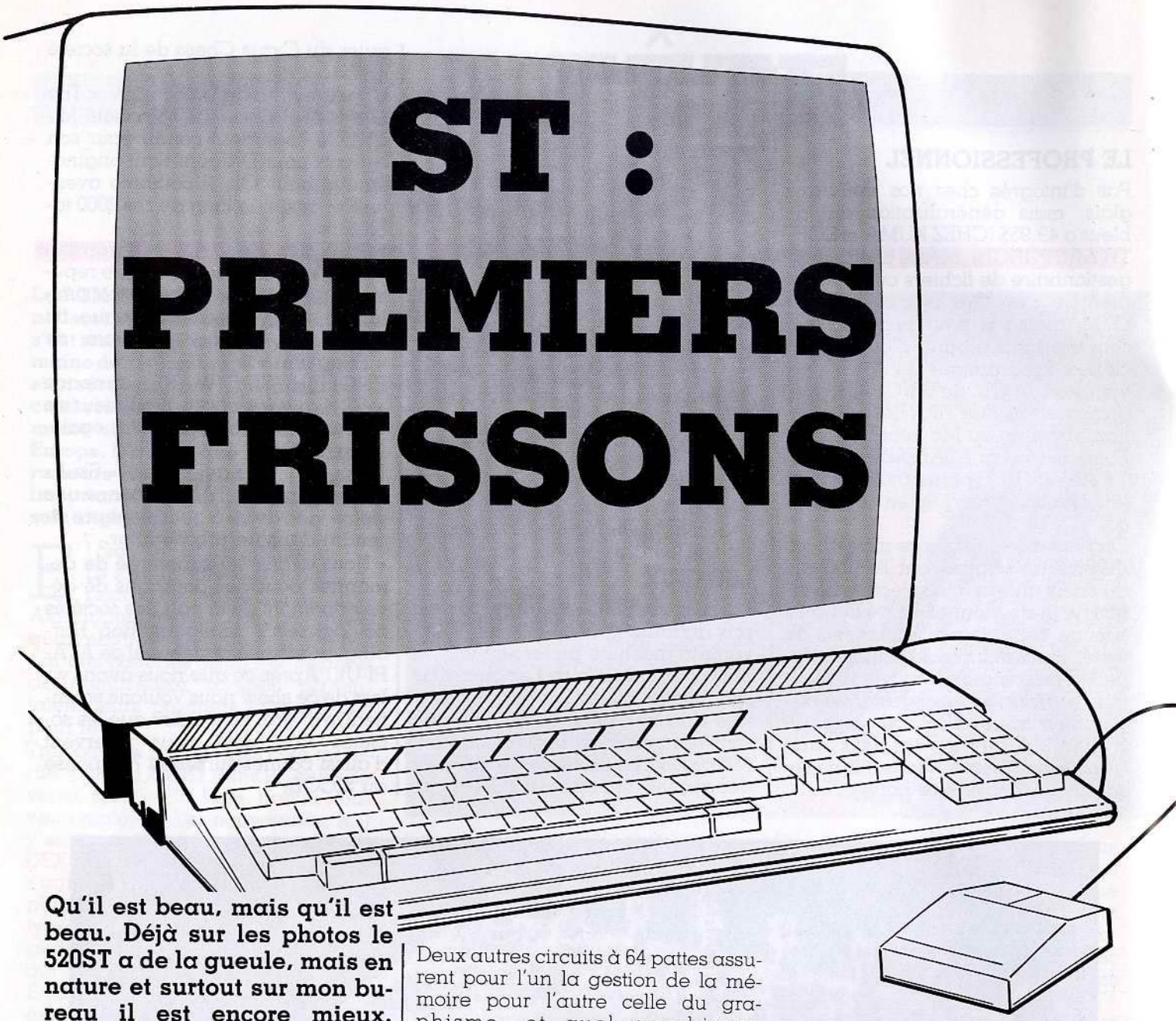
LES JEUX

Par ses possibilités couleurs, par son prix abordable, le 520ST va vite devenir la machine professionnelle la plus pourvue en jeux. Le nombre de simulateurs de vol, de jeux d'aventure et même de jeux d'arcade qui sont annoncés est impressionnant. Même les jeux de stratégie affichent des graphismes de rêve. (Cf l'enchiquier du Cyrus Chess de la société ICS).»
Un jeu était déjà disponible, « The Lands of Havoc » de la société MICRODEAL surtout connu pour son support pour Dragon. Ecrit originalement pour QL, il combine aventure et action le long de ses 2000 tableaux.
FIREBIRD, dont le programme « Elite » sur BBC a acquis une réputation enviable prépare un jeu similaire, mi simulation économique, mi guerre de l'espace du nom de « Starglider ».
ENGLISH SOFTWARE annonce « ST Hyperdrive » un simulateur de course automobile, transcrit également du QL.
Dans les jeux d'aventures, « Skul » (photo jointe) de TALENT donne une petite idée de ce à quoi on peut s'attendre dans un proche futur.
« Nous avons été submergé de demandes pour des packages de développement de la part des sociétés de logiciels » soulignait Rob Harding, directeur commercial de ATARI-UK. Après ce que nous avons vu lors de ce show nous voulons volontiers le croire. Est-ce dire que les sociétés françaises nous réservent d'aussi bonnes surprises ? Réponse au SICOB.



En couleur, il est encore plus beau

ST : PREMIERS FRISSONS



Qu'il est beau, mais qu'il est beau. Déjà sur les photos le 520ST a de la gueule, mais en nature et surtout sur mon bureau il est encore mieux. Après une attente cruelle le voila enfin, accompagné de 25 centimètres d'épaisseur de documentation en anglais. Voici donc mes impressions après 2 mois d'utilisation de cette machine.

1 — L'UNITE CENTRALE

Démonté le 520ST est relativement peu rempli grâce à une intégration poussée et l'utilisation au mieux des circuits intégrés. Le microprocesseur 68000 est un joli monstre à 64 pattes, mais ce sont surtout ses caractèristiques qui en font une bête de course (voir tableau 1). Ici il tourne à 8 MHz. A titre de comparaison les 8 bits ATARI sont à 1,79 MHz.

Deux autres circuits à 64 pattes assurent pour l'un la gestion de la mémoire pour l'autre celle du graphisme, et quel graphisme. 640 × 400 en momochrome (25600 points), 320 × 400 en 4 couleurs, 320 × 200 en 16 couleurs. Tout ces modes demandent 32 ko de RAM pour se loger. Les couleurs sont à choisir parmi 512 combinaisons obtenues par dosage à 8 niveaux du rouge, du vert et du bleu ; ainsi pas de table compliquée à retenir, il suffit de traduire ce que l'on voit : pas assez de rouge, trop de bleu, etc.

Les 512 ko de RAM sont contenus dans 16 boitiers, les 192 ko de ROM ne seront mis en place qu'ultérieurement, actuellement il n'y a qu'un bootstrap de 16 ko. L'échange sera fait plus tard pour une somme modique, paraît-il.

Un circuit 68901 est chargé du port série RS232C programmable de 50 à 19200 bauds, il s'occupe aussi des interruptions et des timers.

La partie sonore interne est confiée à un classique AY-3-8910 de GI, générant 3 canaux sonores de 30 Hz à 125 kHz dont les courbes d'enveloppes peuvent être programmées en ADSR (Attack Decay Sustain Release), en français : Attaque Chute Maintien Relachement. Les deux ports parallèles de ce circuit supportent la sortie imprimante compatible Centronics et divers signaux.

Si le générateur de son date un peu sur une machine de ce niveau, GI en fait de plus puissants, il est largement compensé par la présence de l'interface MIDI, à ma connaissance incorporé pour la première fois en standard dans un micro. En bref, tout est dans le sigle : MIDI = Musical Instruments Digital Interface. Ce qui signifie que le ST est capable d'échanger des ordres musicaux codés

avec des orgues, synthés, boites à rythmes, etc respectant la même norme. Seize appareils peuvent être gérés simultanément. Les ordres adressés au choix à un des instruments, à plusieurs ou à tous ne sont pas des sons, mais bien la manière dont il ont été obtenus : hauteur dans la gamme, toucher, etc. Physiquement il s'agit d'un port série à 31250 bauds confié à un 6850 Motorola.
Le contrôleur de disquette est on ne peut plus connu (FD-1771 de Western Digital). L'interface disque dur travaille en DMA (accès direct mémoire) à une vitesse de transfert de 1,33 Mo/seconde, soit la mémoire totale en moins d'une demie seconde.

TABLEAU 1 LE 68000 de Motorola Bus de données de 16 bits. Bus d'adresse de 24 bits soit 16 Mo de mémoire adressable. 8 registres de données de 32 bits. 9 registres d'adresse de 32 bits. 7 niveaux d'interruptions. 56 instructions dont la multiplication et la division. 14 modes d'adressage. 5 types de données.

— LE CLAVIER

Le clavier est de type dit intelligent, car il est possible de programmer son mode de fonctionnement. Il est piloté par un microprocesseur 6301 et a la charge du clavier AZERTY accentué de 58 touches, du pavé curseur et fonctions spéciales de 8 touches, du pavé numérique avec symboles arithmétiques, parenthèses et retour chariot de 18 touches, des 10 touches de fonctions, de l'horloge temps réel non sauvegardée et pour terminer de la souris et des joysticks dont les mouvements se traduisent par des séquences de caractères. Les touches shift, control et alternate sont utilisées pour générer jusqu'à quatre caractères sur certaines touches. Certains nécessitent l'appui simultané sur shift, alternate et une touche ce qui oblige à lâcher la souris. En revanche le déplacement du curseur se commande d'un doigt. Le toucher est agréable, sans bruit de ferraille ou de plastique creux, et quand vous saurez que pour régler la répétition et le témoin sonore il suffit de prendre la souris et de cliquer...

Bien qu'incorporé dans le même boitier, ce clavier est complètement indépendant de l'unité centrale, il y est simplement relié par le port série d'un 6850 et l'alimentation.

3 — LES CONNEXIONS

Inutile de courir chez le revendeur de pièces détachées du coin pour trouver certaines des prises dont est abondamment pourvu le ST, il ne les a pas. Heureusement d'autres sont standards. Passons les toutes en revue en partant du cotè gauche.
Le port cartouche est classique 2 × 20 broches. Les ports disque dur, série et imprimante aussi, ce sont des Cannon 19 et 25 broches, bien qu'inversées mâles et femelles par rapport à l'usage (j'ai dû souder vite fait des adaptateurs pour utiliser mes cordons). Intercalée, se trouve la prise disquette pas classique du tout. Quatorze broches dans une prise ronde plus grosse qu'une prise DIN. La prise moniteur est loin d'être classique, treize broches sur quatre rangées dans l'encombrement d'une prise DIN. Le signal vidéo couleur, les signaux RVB, le signal TTL monochrome, les synchros, le test du moniteur (le ST vérifie si le moniteur est NB ou couleur) et le son, voilà tout ce qui passe dedans. Donc pour se brancher, il faudra passer par les cordons du constructeur. Autre constatation, il n'est pas possible d'avoir simultanément un écran NB et un couleur.
La sortie télévision aperçue sur les photos des premiers modèles a disparu. L'examen du circuit révèle que l'emplacement du modulateur est actuellement libre sur le circuit imprimé et que les trous sont prêts dans la carrosserie. Renseignement pris chez ATARI-FRANCE, le modulateur sera peut-être monté dans les versions avec ROM complète.
Les deux prises MIDI et la prise alimentation sont à la norme DIN et de modèles courants. Pour terminer, deux prises joysticks au standard ATARI, la première recevant la souris. Notons quand même qu'en dehors de la souris, toute erreur de branchement est écartée par la différence de forme de chaque prise et la gravure d'un symbole doublée de la désignation en clair. Que demander de mieux.

4 — MEMOIRES DE MASSE

En ce qui concerne les mémoires de masse, ici pas de magnéto cassette. Rien que du disque 3,5 pouces, souple simple face 360 ko, double face 720 ko, dur 10 Mo, etc... le CDROM. Encore une nouveauté, un lecteur de disque compact laser de capacité 550 Mo. il n'y a pas de coquille, 550 Mo en lecture seulement. Rapidité de recherche PHE-NO-ME-NA-LE. Prix ? disponibilité ? Wait and see.

5 — LES PERIPHERIQUES

Il est possible de raccorder n'importe quelle imprimante munie d'une interface Centronics ou série.Pas de manœuvre compliquée pour le faire savoir au ST, quelques clics avec la souris. Pas d'imprimante spéciale pour recopier l'écran, ça marche avec les Epson et compatibles. Elles sont légion. Les imprimantes ATARI ne sont pas encore sorties, trois modèles étaient prévus : une à marguerite, une matricielle à impact et une couleur matricielle non impact. Je n'en sais pas plus.
Pour se brancher sur les réseaux pas de problèmes particuliers, l'interface est à la norme RS232, il suffit d'un cordon et d'un moden. Dans le logiciel de base figure un émulateur VI52 et un programme de transmission Kermit. J'ai rencontré un développeur sur ST qui a dans ses cartons un modem 300-300, 1200-75 et 75-1200 symétrisé à un prix raisonnable. Je vous en reparlerai.
Pour voir les belles images produites par l'engin, deux moniteurs proposés par ATARI. Un monochrome pour la haute résolution dont la définition laisse pantois, un point est carré, c'est un point pas une touche. L'image étant rafraîchie 70 fois par seconde, il n'y a pas de scintillement visible. De plus, la déformation est quasi nulle et la stabilité excellente. Il y a des écrans dits pros qui vont être jaloux. Le moniteur couleur, est de même facture, les couleurs sont bien saturées, l'image stable et définie. Le branchement en RVB sur une télé familiale donne d'excellents résultats, le plus difficile étant de bricoler une prise. Gageons que ce problème sera rapidement résolu par quelques boutiques. L'affichage en couleur en

suite page 39

LES 150 PREMIERS

L'Europe, et plus particulièrement l'Angleterre, mêne la danse dans la course au logiciel à destination du ST. Les Etats-Unis, dont provient, pourtant la machine, sont (provisoirement ?) en retrait. Cette explosion, étonnante pour une machine qui n'est disponible que depuis un mois, s'explique de plusieurs façons. En premier lieu, bien sûr, les espoirs de vente de la machine. Mais surtout les différentes compatibilités avec d'autres systèmes.
En premier lieu, TOS est un système d'exploitation très proche de CP/M 68 K.
Ensuite, GEM est disponible sur IBM, APRICOT, TANDY, PHILIPS et autres.
Beaucoup des premiers programmes disponibles sur ST ont été développés sur IBM sous GEM.
Le cousinage fort marqué de l'écran MacInstosh et de l'écran ST est aussi un point positif. L'expérience accumulée par les programmeurs servira pour l'édition sur ST. A tel point que certains éditeurs annoncent la sortie combinée de leurs produits sur Mac et sur ST. Un émulateur McIntosh sur ST fait d'ailleurs l'objet de rumeurs insistantes.
Enfin, la possibilité de faire fonctionner le ST sous d'autres environnements que TOS, BOS par exemple, lui ouvre de nouvelles bibliothèques.

CODE DE PRIX :
A < 500 F
B < 1000 F
C < 1500 F
D < 2000 F
E > 2000 F
F > 5000 F

PROFESSIONNEL

Intégrés :
Le premier logiciel intégré pour le ST sera vraisemblablement INTEGRATED SEVEN de Mosaïc. Ce logiciel comprend un traitement de texte avec correcteur d'orthographe, un tableur, un module graphique, une base de données, un logiciel de communication et un émulateur de terminal. Il offre également un langage de programmation.
Le premier produit GEM de BATTERIES INCLUDED est destiné au 520 ST et à l'IBM PC. Il s'appelle IS (Integrated Software) et comprend un traitement de texte, un tableur avec graphiques, une base de données et un logiciel boursier. Ces programmes travaillent ensemble, mais ne constituent pas un véritable intégré. Le premier module sera disponible cet hiver.
RISING STAR, la société responsable de Valdocs, un des logiciels les plus connus pour sa facilité d'emploi, va éditer 14 modules professionnels pour le ST ainsi qu'un module intégrateur.

Tableurs :

VIP Professional	VIP Technologies	C	Dispo
Clône de Lotus 1.2.3., il sait relire ses fichiers. Moins de 1200 F			
HABACALC'GRAPH	HABA SYSTEM	B	Oct
A environ 600 F, le moins cher des tableurs annoncés.			
THE TWIN	MOSAIC		Nov
Autre clône de Lotus.			
K-SPREAD	SOFT PUNCH	B	Oct
Un tableur est également prévu chez BATTERIES INCLUDED et chez RISING STAR (Voir Intégrés).			

Base de données/Gestion de fichiers :

ST BASE	MICRO APPLICATION	?	Sept
DB MASTER	ATARI	?	Nov
HABAFILE	HABA	B	Sept
DATAMATE	FYDLER	B	Sept
DATASTATE 2	FYDLER	F	Nov
K-DATA	KUMA	B	Sept
SUPERBASE	PRECISION SOFT.	C	Jan
FLEXFILE	TALENT	B	Déc
ST LION	CSD Inc.		Fév. 86
DATABASE	SOFT PUNCH	B	Oct
(Voir aussi Intégrés)			

LOGICIELS

Traitement de texte :

GEM WRITE	DR	Fourni	Sept
EXPRESS Letter Processor	MIRAGE	B	Dispo
Orienté secrétariat avec son fichier d'adresse intégré. Voir Banc d'essai.			
MULTI-MATE	MULTI-MATE		Déc
Puissant, MM offre des possibilités de calcul, une bibliothèque de phrases, des macros commandes et un dictionnaire.			
HABAWORD	HABA	B	Sept
SUPERSCRIPT	PRECISION	B	Jan
K-WORD	KUMA	B	Sept
WORDMATE	FYDLER	B	Sept
SPELLMATE	FYDLER	B	Oct
MAILMATE	FYDLER	B	Oct
ATARI WRITER	ATARI		Déc
WORDPROCESSOR	SOFT PUNCH	B	Sept
(Voir aussi Intégrés)			

Communication :

HABACOM	HABA	B	Sept
MODEM PROGRAM	SST System		Oct
MI TERM	MICHTRON	B	Oct
HABADIALER	HABA		Sept
OMEGA	MICROBITS		
HABAWINDOW DIALER	HABA	Sept	
K-COM	KUMA	B	Sept
PC Intercom-VT100 Emul	MARK OF THE UNICORN	C	Dispo

Réseau

ATARINET	SOFTWARE PUNCH	D	Oct
Permet à 250 utilisateurs de communiquer en réseau local.			

Comptabilité

TWO DEY ACCOUNTING	ATARI	B	Nov
Les sociétés CHIPSOFT, EXECON, QUEST et BOS annoncent des comptabilités complètes en Angleterre. Bien sûr, elles ne sont pas exploitables telles quelles en France.			

Logiciels verticaux

NYMPHIR	COGITAL		Déc
Exploitation d'un cabinet d'infirmiers			
CASH TRADER	QUEST	C	Oct
LOGICIELS développés sous BOS			

Gestion diverse :

HABA CHECK MINDER	HABA	A	Dispo
HOME ACCOUNTANT	MICRO ED INC		Oct
HABATEX	HABA		Sept
HABAWINDOW CALCULATOR	HABA		Sept
ST BUSINESS TOOLS	ATARI		Nov
TIMELINK	SOFTECHNIES	B	Oct
RHYTHM	SOFTECHNIES	A	Oct
NOTEBOOK	SOFT PUNCH	B	Nov
APPOINTMASTER	FYDLER	B	Oct
ADMINISTRATOR	FYDLER	F	Oct
DESK DIARY	PARADOX	B	Sept

PROGRAMMATION

Système d'exploitation

BOS DOS	BOS		Oct
BOS, Business Operating System, est un système d'exploitation qui tourne d'ores et déjà sur le ST. Il permet le fonctionnement de plus de 80 programmes professionnels anglais, qui avant le ST, était disponible sur IBM et APRICOT. Nous reparlerons plus largement de BOS et des applications dans notre prochain numéro.			

LES 150 PREMIERS LOGICIELS

Langages :			
BASIC	DR	Fourni	Sept
MBASIC (Comp. Microsoft)	PHILON		Nov
CRASIC	PHILON		Oct
BBASIC	BTMSOFT		Oct
Assure une compatibilité avec les programmes sur BBC.			
BASIC COMPILER	ATARI		Oct
LOGO	DR	Fourni	Dispo
C COMPILER	LATTICE	?	?
HIPPO C	HABA	A/B	Sept
C COMPILER	PHILON	?	Oct
C COMPILER	DR	?	Sept
C COMPILER	MANX	?	Déc
C DEV.KIT	METACOMCO	C	Oct
QC COMPILER	GST SYSTEM	B	Oct
C COMPILER	COMPUTER ONE	B	Nov
PASCAL	PHILON	?	Oct
PASCAL	METACOMCO	C	Sept
UCSD PASCAL	TDI		Oct
MODULA-2	TDI		Oct
FORTH	DRAGON GROUP	C	Dispo
FORTH Level 2	DRAGON GROUP	D	Nov
FORTH Accelerator	DRAGON GROUP	B	Sept
ASSEMBLEUR	METACOMCO	B	Sept
MACRO ASSEMBLEUR	GST SYSTEM	A	Oct
K-SEKA	KUMA	B	Sept
ASSEMBLER	OASIS		
COBOL	PHILON	?	Oct
ST COBOL	ATARI	?	Oct
FORTRAN 77	PROSPERO	E	Déc
PROLOG	?	?	Dispo
LISP	METOCOMCO	C	Oct
Utilitaires			
IBM ASCII FILE TRANSFER	MICHTRON	A	Sept
M/DISK (RAM DISK)	MICHTRON	A	Sept
SOFT SPOOL	MICHTRON	A	Sept
IBM PC CROSS DEVELOPMENT SYSTEM	METACOMCO		Oct
CREATIVITE			
Graphisme			
GEM PAINT	DR	Fourni	Sept
GEM DRAW	DR		
DEGAS	BATTERIES INCLUDED		Oct
COLOURSPACE	LLAMASOFT	A	Nov
WHITE LIGHTNING	OASIS	A	Nov
MASTERGRAPH	FYDLER	A	Oct
CAD/CM	RISING STAR		Oct
FLEET STREET EDITOR	MIRRORSOFT	B	Nov
VIDEOWORKS	HAYDEN		Déc
DIGITIZER PACKAGE	HAYDEN		Déc
ARTWORKS GRAPHICS	SOFTWARE PUNCH	B	Nov
Musique			
MUSICWORKS	HAYDEN		Déc
MUSIC SYSTEM	ISLAND LOGIC		Déc
Synthèse vocale			
SMOOTHTALKER	FIRST BYTE		Déc
Ecriture			
NOVEL CONSTRUCTION SET	HAYDEN		Déc
EDUCATION			
Ordinateur			
MAC COACH	ATI		Déc

Sciences

HOME PLANETARIUM	DELTRON		Déc
MATH BLASTER	DAVIDSON ASS.		

Machine à écrire

TYPING TUTOR	ACADEMY SOFT		Sept
MASTERTYPE	SCARBOROUGH		Déc
MAC TYPE	PLANTIR		Déc

Médecine

DIET ST	DATA System Inc.		Déc

Vocabulaire

WORD CHALLENGE	HAYDEN		Déc
WORD ATTACK	DAVIDSON ASS.		

JEUX

Adresse :

MARBLE MADNESS	CROSSBEAM		Oct
HEX	MARK OF UNICORN	A	Dispo
GOLDRUNNER	MICHTRON	A	Oct
MISSION MOUSE	PARADOX	A	Oct
MUD SPIES	MICHTRON	A	Dispo
CENTIPEDE	CROSSBEAM		Oct
BATTLEZONE	CROSSBEAM		Oct
ASTEROIDS	CROSSBEAM		Oct
MISSILE COMMAND	CROSSBEAM		Oct
STAR RAIDERS	CROSSBEAM		Oct
JOUST	RUGBY CIRCLE		
THE GOONIES	DATASOFT		
ZORRO	DATASOFT		
ST HYPERDRIVE	ENGLISH SOFT	A	Nov
STARGLIDER	FIREBIRD	A	Janv 86

Aventures :

Texte

Tous les jeux d'aventures d'Infocom sont prévus en versions 520 ST. Ces jeux, par la richesse de la syntaxe et du vocabulaire compris par l'ordinateur, figurent depuis plusieurs années dans tous les Hit-Parade. A recommander, néanmoins, seulement aux anglicists confirmés.

ZORK 1,2,3	INFOCOM	A	Dispo
WISHBRINGER	INFOCOM	A	Dispo
HITCHIKERS GUIDE TO THE GALAXY	INFOCOM	A	Dispo
DEADLINE	INFOCOM	B	Oct
STARCROSS	INFOCOM	B	Oct
SUSPENDED	INFOCOM	B	Oct
WITNESS	INFOCOM	A	Oct
PLANETFALL	INFOCOM	A	Oct
ENCHANTER	INFOCOM	A	Oct
INFIDEL	INFOCOM	B	Oct
SORCERER	INFOCOM	B	Oct
CUTTHROATS	INFOCOM	A	Oct
SUSPECT	INFOCOM	B	Oct
SEASTALKER	INFOCOM	A	Oct
A MIND FOREVER VOYAGING	INFOCOM	B	Oct

(Premier d'une nouvelle série de jeux qui demandent un minimum de 128 K. Un nouveau système de développement permet une interactivite joueur/scénario encore plus grande et un vocabulaire étendu à 1700 mots.)

WEST/SKUL	TALENT	A	Nov

Graphique

TRANSYLVANIA	PENGUIN		Nov

Role

SUNDOG : FROZEN LEGACY	ACCOLADE		Oct
ULTIMA II	SIERRA		Oct
XYPHUS	PENGUIN		Déc

Animé

LANDS OF HAVOC	MICRODEAL	A	Dispo

LES 150 PREMIERS LOGICIELS

Simulations :

Microprose a annoncé pour début 86, la disponibilité de ses nombreuses simulations de vol et de combat.

FLIGHT SIMULATOR	MICHTRON	A	Sept
FLIGHT SIMULATOR	MIRRORSOFT		
FLIGHT SIMULATOR	ISLAND LOGIC		
FLIGHT SIMULATOR	SUBLOGIC		
GATO	SIERRA		Déc
JET	SUBLOGIC		Nov

Sim. Economique

FINANCIAL TIME MACHINE INSIGHT			Oct
MILLIONAIRE	BLUE CHIP		Déc
TYCOON	BLUE CHIP		Déc
BARON	BLUE CHIP		Déc

Stratégie :

CHESS	ICS/PSION		Oct
SARGON III	HAYDEN		Déc
SCRABBLE	HAYDEN		Déc
PENSATE	PENGUIN		Sept
FLIP SIDE (Othello)	MICHTRON	A	Sept
CHIFFRES ET LETTRES	LOGISOFT	A	Oct

Français

LE LIVRE DU 520 ST	MICRO APPLICATION	129 F	Dispo
LE LIVRE DU LANGAGE MACHINE DE L'ATARI ST		149 F	Oct
LE LIVRE DU GEM SUR ATARI ST		149 F	Oct
LA BIBLE DE L'ATARI ST		249 F	Nov
ATARI TRUCS ET ASTUCES		149 F	Nov

En Anglais

THE ATARI ST EXPLORED	KUMA	140 F	Sept
ATARI ST COMPANION	SUNSHINE	150 F	Dispo
ATARI ST BASIC Book 1	GLENTOP	100 F	Sept
WORKING WITH THE ATARI ST	SUNSHINE	150 F	Oct
USING ST LOGO ON ATARI	GLENTOP	140 F	Oct
BEGINNERS 68000 ASSEMBLY ST (avec disque)	GLENTOP	250 F	Oct
PROGRAMMING THE ATARI ST (Vol 1)	COMPUTE	300 F	Nov
ADVANCED USER'S GUIDE FOR ATARI	GLENTOP		Nov
USING GRAPHICS ON ATARI ST	GLENTOP		Nov
BUSINESS APPLICATIONS ON ST	GLENTOP		Nov
ATARI ST USER'S GUIDE	McGRAW HILL	220 F	Déc
COMPUTE'S 1st BOOK OF ATARI ST	COMPUTE	200 F	Déc
USER'S GUIDE TO GEM § TOS	COMPUTE	200 F	Déc
PROGRAMMING THE ST - ADVANCED	McGRAW HILL	260 F	Jan
ELEMENTARY ATARI ST	COMPUTE	240 F	Jan
MAPPING THE ST	COMPUTE	300 F	Fév

En outre les sociétés anglaises BEYOND, BOSDEN, COMPUTER CONCEPTS, COMPUTER ONE, ECCELESTONE, HISOFT, MARTECH, MIRACLE, SOFTWORKS, TYNESOFT et WATSON ont annoncé des développments sur ST. La société américaine SPINNAKER et les sociétés françaises MICRO-APPLICATION, LORICIELS, HATIER, MICROPUCE, COBRASOFT, NORSOFT, MICRO-VIDEO, LOGISPORT et... ATARI préparent également des programmes.

Des précisions supplémentaires sur tous ces softs, des bancs d'essai de ceux déjà parus dans notre prochain numéro.

..... RUMEURS ..... RUMEURS.....

— Un wargame est annoncé chez Miles Computing, il s'agirait de la version ST de Mac Attack.

— HABA dont un certain nombre de produits figurent dans la liste logiciel de ce numéro convertirait désormais tous ses produits MacIntosh pour le 520 ST.

— DIGITAL RESEARCH annonce une liste de 25 sociétés de soft supportant des applications sous GEM.

— C'est ainsi que ASHTON TATE, bien que n'ayant pas acheté de système de dévelopement ST, pourrait éditer DBASE II/ DBASE III et FRAMEWORK pour le 520 à partir des versions GEM qu'il prépare. — C'est le cas aussi de SORCIM qui prépare SUPERCALC 3 et SUPERPROJECT, de Macthinck Corp qui prépare ST THINK, et CHANG LABS éditeur de RAGS TO RICHES.

— En revanche malgré de nombreuses rumeurs, toujours pas d'annonce officielle de la part de MICROSOFT. C'est, si l'on y réfléchit bien, un peu normal. Dans la troisième manche qui l'oppose à DIGITAL RESEARCH, maître d'œuvre de tout le soft du ST, il n'est sans doute pas souhaitable de donner déjà des armes à son adversaire.

BULLETIN D'ABONNEMENT

NOM _____ ADRESSE COMPLETE _____

_____

Je m'abonne à partir du numéro _____ de ST MAGAZINE. Envoyez votre règlement (Chèque, CCP) à PRESSIMAGE, 210, rue du Fg St Martin, 75010 PARIS

ST Magazine vous apportera durant les prochains mois :
— Des articles d'initiation aux différents langages disponibles sur le ST (Basic, Logo, Langage C, Pascal, Assembleur 68000) et a ses systèmes d'exploitation (TOS, GEM).
— Des programmes gratuits
— Des trucs techniques
Des bancs d'essai et des comparatifs pour vous aider a choisir vos logiciels et vos périphériques.
— Des dossiers complets thématiques pour rafraichir vos connaissances.
— Une rubrique professionnelle régulière pour rentabiliser au mieux l'outil ST dans le cadre de votre travail.
Pour utiliser au mieux la puissance du 520 ST, pour être sûr de ne pas manquer un numéro

ABONNEZ VOUS A ST MAGAZINE 10 Numéros : 200 F.

Au sommaire du prochain numéro:
Le ST chez un graphiste.
Deux tableurs à la loupe.
Comment développer sur ST.
Dossier Imprimante.
et tous les derniers programmes.

ST MAG

ST LANGAGES

L'angoisse de la feuille blanche vous étreint devant l'écran du ST. Le LOGO et le langage C devant être disponibles au moment ou vous lirez ces lignes, nous avons concocté, dans le cahier qui suit, quelques listings pour ceux qui savent, et les premières notions pour ceux qui voudraient bien savoir. Le langage C sera très certainement le langage de choix pour le développement de programmes sur le ST, aussi vous encourageons nous très fortement à vous familiariser avec ce langage.
Le LOGO possède l'extrême avantage d'être disponible et la rareté des autres langages nous l'a fait redécouvrir. Avec nous, faites vos premiers pas. Vous découvrirez en prime les premiers gribouillis de votre ordinateur. Il a du talent, le bougre !
Nous n'oublierons pas, dans nos prochains numéros de vous parler du Basic et de l'Assembleur 68000. Et peut être même de Prolog. Patience !

IMPRIMER UN FICHIER

```
/*      Programme pour imprimer un fichier avec quelques commandes de   */
/*      formatage. Le programme demande le nom du fichier a utiliser    */
/*      et envoie certaines commandes a l'imprimante.                   */

#define OUVRIR          (char)  0x3d            /* gemdos codes         */
#define LIRE            (char)  0x3f
#define FERMER          (char)  0x3e
#define IMPRIMANTE      (char)  0x05
#define ECRAN           (char)  0x02

extern long gemdos();                           /* la fonction systeme  */

static int nrofichier;                          /* N.de lien du fichier */

#include <stdio.h>
/* ATTENTION: stdio.h definit un MACRO avec le nom max (en minuscules)  */
/* "max" est alors un mot RESERVE!, et si vous definissez un char max;  */
/* par exemple, vous pouvez passer une demi-journee a chercher le bug!  */

main()
{
        char nomfic[30];        /* Nom du fichier a imprimer            */
        int essais;             /* No des essais pour ouvrir le fichier */

        essais = 0;
        titre();                /* Affiche le titre du programme        */
        getfic(nomfic);         /* Met le nom fourni par l'utilisateur  */
                                /* dans la chaine "nomfic"              */
        while ((ouvrirfic(nomfic) < 0) && (essais < 2)) essais++;
        if (essais == 2) return(-1);    /* rate!                        */
        initprt();              /* Initialisation de l'imprimante       */
        print();                /* Envoie la sauce                      */
        close();                /* ferme tout et salut!                 */
}

titre()
{       /* affiche le titre     */
        putstring(ECRAN,"Programme d\'impression\n\r\0");
        putstring(ECRAN,"\n\r\tpar jakob navia 4 sep. 85\n\r\0");
                        /* votre nom ici s.v.p...                       */
}

initprt()
{                       /* Initialisation de l'imprimante               */

        putstring(IMPRIMANTE,"\033\100\r\0");   /* Initialisation       */
        if (ouinon("Marges (haut=4, bas=4)?\0"))
                {                               /* ESC "R" 4 (haut)     */
                putstring(IMPRIMANTE,"\033R\004\033N\004\r\0");
                        /* CHR$(27,"R",4) en Basic      */
                } else putstring(IMPRIMANTE,"\0330\r\0");
        if (ouinon("double impression?\0"))
                {
                putstring(IMPRIMANTE,"\033B\004\r\0"); /* ESC B chr$(4) */
                }
        ouinon("Imprimante EN LIGNE?\r\n\0");
}

getfic(nomfic)
char nomfic[];
```

C

```
{
        int OKFLAG;

        OKFLAG = 1;
        while (OKFLAG)
        {
        putstring(ECRAN,"\n\nNom du fichier?\0");
        getstring(nomfic,(int)20);
        putstring(ECRAN,"\n\rFICHIER \0");
        putstring(ECRAN,nomfic);
        OKFLAG = !ouinon("OK?\0");
        }
}

ouinon(str)             /* Rend 1 si oui, 0 si autre chose      */
char *str;
{
        char t;

        putstring(ECRAN,"\n\r\0");
        putstring(ECRAN,str);
        putstring(ECRAN," OUI ou NON? (o/n) \0");
        t = getchar();
        if ((t=='o')||(t=='O')) return(1); else return(0);
}


ouvrirfic(nomfic)       /* Ouvre le fichier indique par nomfic  */
char nomfic[];          /* Si c'est reussi rend 1 si non -1     */
{

        nrofichier = (int)gemdos(OUVRIR,nomfic,(int)0);
        if (nrofichier < 0)
                {
                putstring(ECRAN,"ERREUR: Impossible d'ouvrir le fichier \0");
                putstring(ECRAN,nomfic);
                putstring(ECRAN,"\n\rpressez return pour continuer...\0");
                getchar();
                return(-1);
                }
}
getstring(str,maxchar)  /* Rend une chaine de maximum maxchar caracteres */
char str[];
int maxchar;
{
        char t[80],x;
        int i;

        t[0] = (char) maxchar;
        gemdos(0x0A,t); /* gemdos fonction pour lire une chaine: le     */
                        /* premier caractere est la longueur maximum    */
                        /* du tampon, et le deuxieme la longueur reelle */
        for(i=2;(int)t[i];i++)  str[i-2] = t[i];/* Ici on supprime      */
        str[i-1] = '\0';                        /* les 2 premiers car.  */
}

putstring(ou,chaine)    /* Impression d'une chaine ou a l'ecran avec    */
char ou, chaine[];      /* putchar,ou a l'imprimante avec appel systeme */
{
        char t;

        while(t = *chaine++)
                if (ou==ECRAN) putchar(t); else gemdos(IMPRIMANTE,t);
}
```

```
print()                 /* Envoie le fichier a l'imprimante ET a l'ecran */
{
        char tampon[156];
        int n;

        n = 155;
        while (n == 155)
        {
        n = gemdos(LIRE,nrofichier,(long)155,tampon);
        tampon[155] = (char)0;
        putstring(ECRAN,tampon);
        while (!gemdos(0x11));  /* attend l'imprimante...       */
        putstring(IMPRIMANTE,tampon);
        }
        putstring(IMPRIMANTE,"\r\f"); /* Ne pas oublier de vider le buffer */
}

close()
{
        gemdos(FERMER); /* Fermeture du fichier         */
}
/* ouf!...      */}
        putstring(IMPRIMANTE,"\r\f"); /* Ne pas oublier
```

QU'EST CE QUE « C »

Le mythe du langage structure

On a beaucoup entendu des phrases du type: "C est un langage structure, il est mieux que Basic" ou encore: "Le langage C empeche l'ecriture du code du type spaghetti" etc.

Beaucoup de gens qui aiment repeter ce qu'eux-memes ne comprennent pas, repetent cette phrase (ou des phrases similaires) a satiete.

Un langage, qu'il soit un langage informatique ou un langage humain, est d'abord une NOTATION pour exprimer un certain nombre de choses. Dans le cas concret du langage informatique, il decrit une serie d'actions a realiser par l'ordinateur. Avec le meme langage on peut s'exprimer d'une facon claire ou d'une facon completement illisible. La clarte est plutot une propriete des gens, une caracteristique de leur pensee (et aussi evidemment elle depend de la complexite du sujet a traiter...), qu'une caracteristique du langage meme. On peut ecrire des codes du type spaghetti dans n'importe quel langage informatique, et "C" n'est pas une exception, surtout si on pense que le fameux GOTO est utilisable dans "C" (heureusement...)

Les vrais avantages de "C"

A mon avis, "C" a plusieurs avantages:
1. Les appels a des subroutines par leur nom.
2. Les variables locales.
3. Les donnees structurees.
4. La structure en blocs.

Les appels aux subroutines par leur nom.

Imaginons la sequence: (Basic)

```
100 A = 128
110 B = 345
120 GOSUB 6850
```

La meme chose devient beaucoup plus claire en "C":

```
point(128,345);
```

Cette lacune grave du Basic nous donne beaucoup de fil a retordre quand on essaie de comprendre ce que le programme que nous avons devant les yeux doit faire (et pourquoi, helas, il ne le fait pas!).

Les variables locales

Je ne crois pas qu'un seul programmeur en Basic n'ait pas connu le bug suivant:

```
180 FOR I = 1 TO 10
190 A = 50 * I
200 B = 60 / I
210 GOSUB 5400
220 NEXT I
```

Cette routine ne fonctionne pas. Pourquoi? En regardant de pres la routine a la ligne 5400 on voit qu'elle fait:

```
5400 FOR I = 25 TO 35
5450 ... traitement ....
5490 NEXT I
5600 RETURN
```

La valeur de I a ete modifiee par la subroutine a la ligne 5400 et la boucle principale n'a ete effectuee qu'une seule fois car la valeur de I etait 35 au retour de la ligne 5500. On peut eviter ce piege en BASIC si on utilise systematiquement une numerotation de variables du style X1,X2,X3,... X345, etc C'est un procede tres lourd et pas toujours efficace.

En "C", c'est simple: toutes les variables declarees au-dedans d'une subroutine sont "LOCALES" par rapport a la subroutine d'appel, et ne sont pas connues en dehors. Alors on peut toujours utiliser la variable I comme compteur de boucles sans se tromper.

Les donnees structurees.

En BASIC une variable peut etre ou numerique ou alphabetique. Il n'est pas possible de definir une variable "point" par exemple, avec 2 valeurs: coordonnee x, et coordonnee y. Pour mettre en evidence qu'il s'agit du meme point on peut en BASIC utiliser des noms complexes (du style "POINTORIGINE") mais c'est completement artificiel et etranger a la structure du langage. En "C" par contre on peut definir une "structure" de donnees du type point:

```
struct point
        {
        int xcoord, ycoord;
        };
```

alors on peut utiliser simplement:

```
point.xcoord = xdepart;
point.ycoord = ydepart;
```

On peut aussi de cette facon donner une structure comme argument a une routine:

```
ligne(point1,poin2);
```

La structure des fonctions

Les fonctions permettent de diviser la tache a realiser de facon a ecrire de petits programmes. Cela clarifie l'ensemble du programme et permet de developper des modules a utiliser sans modifications dans plusieurs programmes.

Chaque fonction a la forme:

```
[type de resultat] [NOM] ( liste d'arguments )
declarations d'arguments;
{
declarations des variables locales
...
corps de la fonction
}
```

Exemple:

```
double atod(chaine)
char *chaine;
/* Fonction pour lire une chaine de caracteres (pointee par "chaine") qui   */
/* renvoie un nombre dans le format "double" (64 bits).                     */
/* La declaration de la fonction specifie un resultat de type "double"      */
/* et declare l'argument "chaine" comme etant un pointeur sur des caracteres */
/* Retourne -1 si elle rencontre un caractere different de 0...9            */
/* Ne prend pas en compte le signe!                                          */

{       /* debut du "atod", ascii to double     */
        double i;

        i = 0.0;            /* Initialisation de i. Notez 0.0 et non 0!     */
        while ( *chaine >= '0' && *chaine <= '9') /* chiffres 0-9 seulement  */
                {
                i = 10.0 * i + (*chaine - '0') /* additione a i la valeur de */
                                               /* chaque chiffre             */
                chaine++;                      /* augmente le pointeur       */
                }
        if (*chaine) return (-1.0);            /* voir les explications...   */
        return(i);
}       /* fin de atod  */
```

Remarques:

1. La fonction atod est declaree comme une fonction qui retourne un double double atod(chaine)
2. L'argument chaine est declare comme un pointeur. Voir plus loin.
3. La variable i est declaree comme une variable locale a atod (double i;)
4. L'expression && est equivalente a la fonction AND du BASIC.
5. L'expression *chaine - '0' est equivalente a VAL(A$).
6. Le test if (*chaine) teste si le pointeur est a zero. S'il n'est pas a zero, cela implique que le programme a rencontre un caractere non numerique. (Voir le test au debut du while). Alors on renvoie un chiffre negatif pour signaler l'erreur au programme d'appel. Sinon on renvoie le resultat i.
7. Les commentaires son encadres par /* (debut du commentaire) et */ (fin du commentaire).

Les structures iteratives (boucles...)

En "C" il existe plusieurs structures pour realiser une boucle:

La boucle "for".
Syntaxe:

```
for (i=0 ; i<10 ; i++)
        {
        ... traitement ...
        }
```

La meme chose en BASIC:

```
100 FOR I = 0 TO 9 STEP 1
110 ... traitement ...
200 NEXT I
```

"C" execute avant la boucle la premiere partie de l'instruction. Dans notre exemple, il initialise i a la valeur 0. Apres, "C" teste la condition, dans le cas cite si i est inferieur a 10. Si la condition est fausse, "C" saute toute la boucle, comme presque tous les BASICs. A chaque boucle, "C" execute les operations specifiees dans la troisieme partie de la boucle: ici il augmente de 1 la valeur de i.

Les instructions BASIC:

```
100 FOR I = 21 TO 2.3 STEP -0.45
110 ... traitement ...
120 NEXT I
```

correspondent en "C" a:

```
for (i=21.0 ; i >= 2.3 ; i = i - 0.45)
        {
        .. traitement ...
        }
```

Une boucle infinie est facile a ecrire:

```
for( ; ; ;)
        {
        .. traitement ...
        }
```

Si on veut executer plusieurs instructions a la fin de chaque iteration, on peut les ecrire dans la troisieme partie de l'instruction for, separees par des virgules:

```
for (i=0 ; i <= 10 ; i++ , f = f - 3 , z = 0)
        {
        ... traitement ...
        }
```

A chaque iteration, i sera augmentee de 1, f diminuee de 3, et z remise a zero.

Les boucles "WHILE".

La syntaxe de while est:

```
while (condition)
        {
        ... traitement ...
        }
```

Le programme execute le traitement TANT QUE (while) la condition est vraie.

Les boucles do-while

Les boucles for et while testent si la condition est vraie avant de commencer l'execution du corps de la boucle. L'instruction do while teste a la FIN si la condition est vraie, et alors la boucle sera executee toujours au moins 1 fois.
Syntaxe:

```
do
        {
        ... traitement ...
        } while (condition);
```

L'instruction "BREAK" et l'instruction "CONTINUE".

L'instruction break fournit un moyen de sortir d'une boucle et de continuer le traitement comme si la condition de la boucle etait fausse sans utiliser trop les GOTOS.
L'instruction "continue" permet d'executer l'iteration suivante immediatement.

L'instruction "SWITCH" (cas)

Cette instruction permet d'eviter d'innombrables IF (condition) GOTO avec un format tres lisible: Voila un exemple pour le cas tres classique d'un menu avec plusieurs options numerotees de 1 a 4. D'abord on ecrit l'exemple BASIC:

```
100 PRINT "Rentrez un chiffre de 1 a 10"
110 INPUT A$
120 IF A$ = "1" GOSUB 5500
130 IF (A$ = "1") OR (A$ = "2") GOSUB 6000
140 IF (A$ = "1") OR (A$ = "2") or (A$="3") GOSUB 6500
150 IF (A$ = "1") OR (A$ = "2") OR (A$="3")
    OR (A$="4") GOSUB 7000
160 GOTO 190
170 PRINT "ERREUR"
180 GOTO 100
190 REM Continuation du programme...
```

Avec "C", la meme construction est tres simple:

```
etiquette: printf("Rentrez un chiffre de 1 a 10\n");
c = getchar();
switch (C)
        {
        case "1": traitement1();
        case "2": traitement2();
        case "3": {
                  traitement3();
                  break;
                  }
        default:  {
                  printf("Erreur.\n");
                  goto commence;
                  }
        }
```

Attention: Avec "C" si le premier cas est vrai (la reponse est "1") le programme va executer traitement1, ET traitement2, ET traitement3 . C'est pour simuler cette caracteristique qu'on est oblige d'utiliser GOSUBs et non pas GOTOs dans la version Basic.

Si on veut eviter cela, il faut rajouter une instruction "break" (voir les explications plus loin) qui interrompt l'execution du bloc courant. Alors aura:

```
switch (c)
{
case "1": traitement1(); break;
case "2": traitement2(); break;
case "3": traitement3(); break;
default: printf("erreur\n");
}
```

L'instruction IF ELSE

Presque pareil au IF THEN ELSE de certains BASICs: la syntaxe est la suivante:

```
if (condition) instruction1; else instruction2;
```

si (condition) alors execute instruction1, sinon execute instruction2

Exemple:

```
if (x == 45) z += 4; else y += 4;
```

si x est egal a 45 alors on ajoute 4 a z, sinon on ajoute 4 a y. Notez que la comparaison de x avec 45 est ecrite comme ==.

Remarques:

==> Si l'expression dans (condition)est differente de zero elle est consideree comme vraie.

==> Else s'applique toujours au dernier if dans les cas de plusieurs ifs emboites.

==> Notez que toutes les instructions finissent par un point virgule, au contraire du Pascal, qui exige d'eliminer le point virgule avant l'else.

Les types predefinis

Il y a 5 types de donnees de base en "C":

1. char
   En general c'est la meme chose qu'un byte.
2. short
   Entier sur 2 bytes ou 16 bits.
3. int
   Entier sur 4 bytes ou 32 bits.
4. float
   Nombre en virgule flottante avec 32 bits.
5. double
   Nombre en virgule flottante avec 64 bits.

Le compilateur doit tenir compte des transformations necessaires et les effectuer lorsqu'il rencontre des expressions telles que:

```
c = a / b;
```

Si a = int, et b est un double, le compilateur transforme a en double, et effectue la division apres. Le resultat est du type double. L'utilisateur peut aussi "forcer la main" et indiquer explicitement les types de conversions souhaitees avec l'operation "cast":

```
(double) p
```

Ceci indique au compilateur que le nombre qui suit (p) devrait etre transforme dans le format double, c'est-a-dire 64 bits.
Attention: la variable p reste intacte, c'est seulement une operation de type temporaire. Si on veut sauvegarder le resultat d'un "cast", il faut le faire dans une autre variable, qui a ete declaree comme double:

```
f = (double) p;
```

La valeur en format double de p est sauvegardee dans f.

Les constantes.

Il y a differentes facons de representer les constantes:

1. Precedees par un 0: constantes en base huit (octal).
   Exemple: 040 (32 decimal)
2. Suite de chiffres. Constantes en base dix.
   Exemple 546
3. Precedees par 0x. Constantes en base seize (hexadecimal)
   Exemple: 0x0a (decimal 10)
4. Suivies d'un l, une constante entiere dans n'importe quelle base sera convertie en long int.
5. Les constantes caracteres sont constituees d'un caractere compris entre apostrophes comme dans 'z'.
   Certains caracteres peuvent etre representes par des symboles:

interligne	NL	\n
tabulation	HT	\t
espacement arriere	BS	\b
retour chariot	CR	\r
saut de page	FF	\f
slash inverse	\	\\
apostrophe	'	\'
configuration de bits	aaa	\aaa

Le code \aaa represente 2 ou 3 chiffres en base huit. Pourquoi en base huit ? Pourquoi faire simple quand vous pouvez faire complique ? Le langage "C" a ses "bugs" (comme tous les langages) et les chiffres en base huit c'est pas le pied...

6. Les constantes en virgule flottante:
   Exemple: 3.141592653, ou 1.34E15
7. Les chaines de caracteres sont separees par des guillemets:
   Exemple: "Je suis une chaine malheureuse..."

Les declarations:

Avant d'utiliser une variable,il est necessaire de la declarer. Il n'est pas possible d'utiliser les variables avant, et beaucoup de critiques du langage "C" sont du type: "C'est tres lourd d'ecrire tant de declarations...". Cependant les declarations sont tres utiles apres, quand on lit le programme et il n'est pas necessaire de suivre tout le programme pour savoir de quoi il s'agit quand on voit la variable toto23 apparaitre dans le code...
BASIC, en plus, permet d'initialiser les variables par defaut a zero ce qui complique encore les choses...
Une declaration est simplement un type deja defini, suivi par les noms d'une ou plusieurs variables. Par exemple:

```
int a,b,c;
double f;
char c;
```

Il est possible d'initialiser les variables lors des declarations:

```
int i = 0;
char c = 'T';
double pi = 3.141592653;
int tableaux[] = { 1, 3 , 5 , 7 };      /* tableaux de 4 elements */
char *s = "voila\0";                    /* chaine de caracteres */
```

Les noms des variables sont constitues de lettres et de chiffres; le premier caractere doit etre une lettre. Le caractere souligne est considere comme une lettre. Alors on peut construire de noms comme:

```
bilan_total
```

pour rendre les choses plus claires. Les caracteres majuscules et minuscules sont differents. La convention en "C" est d'utiliser les majuscules pour les constantes et les minuscules pour les variables.

En "C" il est possible d'additionner sans problemes des nombres entiers avec des caracteres.
Par exemple, si la variable a est un caractere compris entre '0' et '9', l'expression:

```
a - 48
```

donne la valeur numerique de la variable. (48 correspond a la valeur ASCII du '0'). Dans toute expression arithmetique, un caractere est considere comme etant un entier.

Les operateurs d'augmentation et de diminution

En "C" nous avons deux operateurs nouveaux pour augmenter ou diminuer des variables.
L'expression:

```
a++;
```

augmente de 1 la valeur de la variable.
L'expression:

```
b = a++;
```

sauvegarde dans b la valeur de a, et APRES augmente de 1 la valeur de a.
L'expression:

```
b = ++a;
```

D'ABORD augmente la valeur de a, et apres sauvegarde la valeur de a dans b.

De la meme facon, l'operateur -- diminue la valeur de a dans:

```
b = a--;
```

Les operateurs logiques de bits.

Les operateurs logiques nous permettent de travailler avec les bits. Tous les operateurs logiques doivent travailler avec des entiers.

```
&   : ET (Ne pas confondre avec l'operateur unaire &, qui rend
          l'adresse de la variable qui le suit. Ici c'est un
          operateur binaire, i.e. avec 2 arguments.)
|   : OU (inclusif)
^   : OU (exclusif)
<<  : decalage a gauche
>>  : decalage a droite
~   : negation (complement a un. operateur unaire)
```

AND et OR nous etant connus depuis Basic, il faut seulement expliquer les operateurs de decalage:

L'operateurs << , >> , et ~ .

Si la variable a est un entier, et sa representation binaire est :

```
a = 00010011 (19 decimal)
```

avec l'expression:

```
a << 3;
```

on decale de 3 bits vers la gauche, ce qui donne 10011000 ou 128 + 16 + 8 = 152

L'operateur >> est exactement pareil, mais avec la seule difference qu'il decale les bits vers la droite. Ainsi si a est 00010011
l'expression:

```
a >> 3
```

decale de 3 bits vers la droite, ou 00000010 = 2.

L'operateur ~ est tres simple: il convertit tous les 1 en 0 et inversement. Par exemple, si a dans notre exemple vaut 00010011, l'expression:

```
~a
```

vaut 11101100 = 128 + 64 + 32 + 8 + 4 = 236.

Les operateurs et les expressions d'affectation.

Des expressions comme M = M + 8 peuvent etre ecrites de facon plus simple sous la forme M += 8. La plupart des operateurs binaires (operateurs avec 2 arguments, a droite et a gauche: + - * / % << >> & ^ |) ont un operateur d'affectation, c'est-a-dire:

```
M |= 011001;
```

c'est la meme chose que

```
M = M | 011001;
```

La sortie avec format:printf.

Nous avons utilise dans les exemples precedents plusieurs fois la fonction "printf". Cette fonction equivaut au PRINT USING du Basic, cependant avec quelques gadgets en plus:

La syntaxe est tres simple:

```
printf("... arguments..." , donnees);
```

Les arguments sont precedes du "%", et specifient le type de donnees a formater

```
printf(">%d",45);              produit le resultat:>45.
printf(">%d7",45);             produit le resultat:>     45
printf(">%d7.5",0.0545676);    produit le resultat:>0.05456
printf(">%s","chaine\0");      produit le resultat:>chaine
```

Les differents types d'objets et de formats sont les suivants:

1. %d L'argument correspondant doit etre une variable qui pointe sur un entier (d comme decimal)
2. %o Formatage de l'argument en base huit (octal).
3. %x Formatage de l'argument en base seize (hexadecimal)
4. %h L'argument doit etre un short (16 bits).
5. %c Formatage d'un seul caractere.
6. %s Formatage d'une chaine de caracteres. Elle doit se terminer par 0.
7. %f Formatage de donnees en virgule flottante. Optionnellement, on peut donner la longueur totale du format et le nombre de chiffres apres la virgule. %f9.4 c'est le formatage de longueur totale 9 avec 4 chiffres apres la virgule.

Quelques exemples:
1.

```
a = 5;
b = 45.78998
printf("[%d] unites. Moyenne: %f\n",a,b);
```

Resultat:

```
[5] unites. Moyenne: 45.789
```

2.

```
        a = 10;
        printf("octal: %o decimal %d hexadecimal %x\n",a,a);

        Resultat:
octal 12 decimal 10 hexadecimal A
```

Printf permet aussi de formater les donnees dans une chaine de caracteres:

```
        printf(s, "%f",a);
```

Il permet de recuperer dans la chaine de caracteres s le resultat du formatage du chiffre a.
Il est vrai que quelques Basic ameliores arrivent presque avec leur PRINT USING a la flexibilite de printf, mais je ne crois pas qu'un seul Basic puisse bien emuler printf.

Les fonctions similaires sprintf et fprintf ont la meme syntaxe en ce qui concerne les commandes de formatage, mais permettent de sauvegarder le resultat dans une chaine de caracteres (sprintf, string printf) ou dans un fichier (fprintf, ou file printf). La syntaxe est:

```
        sprintf(s, "[ %d ] unites \n",u);
```

La chaine de caracteres s recoit le resultat: "[50] unites \n\0"
(\n est le caractere pour passer a la ligne, et \0 indique que la chaine se termine par un 0, comme il le faut en langage "C").
La syntaxe de fprintf est exactement similaire:

```
        fprintf(fichier, "specs" , donnees);
```

Les pointeurs.

"Un pointeur est une variable qui contient l'adresse d'une autre variable." (Kernighan et Ritchie p. 87) En "C", les adresses de variables ou structures sont des variables tout a fait etablies dans la syntaxe habituelle du langage L'operateur & donne l'adresse de l'objet en question, et l'operateur * donne le contenu de l'adresse qu'on lui fournit.

Les pointeurs peuvent etre les arguments d'un sous-programme, et de cette facon permettent aux subroutines de modifier les arguments du programme d'appel. Au lieu de donner comme arguments les objets a modifier eux-memes, on donne des pointeurs sur ces objets, et on modifie leur valeur avec l'aide de l'operateur *. Evidemment, en Basic on n'a pas ce probleme:toutes les variables sont modifiables par n'importe quel sous-programme, avec les mauvaises consequences qu'on a vues tout a l'heure.

Les tableaux

En "C" la declaration d'un tableau est:

```
        int tab[23];
```

qui definit un tableau de 23 elements, baptises tab[0], tab[1] , tab[2] etc.
Il ne faut pas oublier que la numerotation commence a 0, donc le cinquieme element est tab[4]!
Pour declarer un tableau de 2 dimensions on ecrit:

```
        int grand_tab[3][4];
```

grand_tab est ici un tableau de 3 lignes et 4 colonnes. Pour y acceder a la deuxieme ligne et la 3me colonne on ecrit:

```
        grand_tab[2][3]
```

et NON

```
        grand_tab[2,3]    /* Faux */
```

On peut faire des tableaux de n'importe quel type d'objet, et pas des chiffres seulement.

Resume

Operations binaires: (expression) operation (expression)

* / %	Multiplication, division, et modulo.
+ -	Addition et soustraction.
>> <<	Decalage a droite, decalage a gauche.
< >	Plus petit que, plus grand que.
<= >=	Plus petit ou egal a, plus grand ou egal a.
== !=	Egal a, different de.
&	adresse.
^	ou exclusif
|	ou inclusif
&&	ET
||	OU

Operations d'affectation:

Elles ont toutes la meme priorite et affectent de droite a gauche:

=	affectation
+= , -= , *= , /= , %= , >>= , <<= , &= , ^= , |=	

Instructions:

```
-->     if (expression) instruction;
        if (expression) instruction; else instruction;
-->     while (expression) instruction;
-->     do instruction while expression;
-->     for (expression1;expression2;expression3) instruction;
-->     switch (expression) instruction
                case expression-constante : instruction
        default: instruction;
        break;
        continue;
-->     goto instruction;
-->     return(expression);
        return;
-->     etiquette :
```

Operateurs unaires:

* expression	Valeur contenue dans l'adresse donnee par l'expression.
& expression	Adresse de l'expression.
- expression	Changement de signe de l'expression.
~ expression	Negation de l'expression.
++ valeur	Augmentation avant l'utilisation
valeur++	Augmentation apres l'utilisation
-- valeur	Diminution avant l'utilisation.
valeur --	Diminution apres l'utilisation.
(type) valeur	Cast: Conversion du type de l'expression au type indique par (type)

JACOBO NAVIA

Dans notre prochain numero:

"C" dans l'ATARI 520.
La gestion de votre "Menu",

LOUPE

LOUPE est un programme permettant d'agrandir une partie de l'ecran, ce qui peut etre utile pour étudier les éléments d'un dessin, par exemple...

COMPILATION :

Utiliser la séquence normale de compilation :
- C b:LOUPE
- Puis LINKACC b:LOUPE

Il suffit ensuite de renommer le progamme DESK.ACC

UTILISATION

Selectionner la loupe dans les instruments de bureau :
- déplacer la souris sur l'écran, la partie pointée s'affiche agrandie dans la loupe...

Pour terminer, stabiliser la loupe, et cliquer.
La flèche réapparait, et c'est fini...

```
/*********************************************************************/
/* LOUPE : accessoire de bureau                                      */
/*-------------------------------------------------------------------*/
/*                                                                   */
/*      (c)      1985 par Jacob joel                                 */
/*********************************************************************/

/*********************************************************************/
/* Fichiers a inclure                                                */
/*********************************************************************/

#include "obdefs.h"
#include "define.h"
#include "gemdefs.h"
#include "osbind.h"
```

```
/*********************************************************************/
/* Variables externes                                                */
/*********************************************************************/

extern int      gl_apid;

/*********************************************************************/
/*        VARIABLES                                                  */
/*********************************************************************/

int     gl_hchar;
int     gl_wchar;
int     gl_wbox;
int     gl_hbox;        /* taille des caracteres */

int     pxyar[5];

int     menu_id ;       /* notre menu id */

int     phys_handle;
int     handle;

int     xdesk,ydesk,hdesk,wdesk;  /* taille du bureau */

int     msgbuff[8];     /* buffer des evenements              */
int     keycode;        /* code touche                        */
int     mx,my;          /* souris  x , y pos.                 */
int     butdown;        /* etat du bouton ------   UP/DOWN    */
int     ret;

/* forme de la loupe                                 */

int     loupe_forme[37] = {
        7,7,1,0,1,
        6336,8224,22544,32776,8,8,8,8,16,56,32772,16386,8257,8096,16,8,
        1792,6336,10016,16528,32848,32848,32784,32784,49184,41024,24504,
        8316,8126,31,15,7
        };

/* variables necessaires pour GEM */

int     contrl[12];
int     intin[128];
int     ptsin[128];
int     intout[128];
int     ptsout[128];

int work_in[11];
int work_out[57];
int pxyarray[10];

/*********************************************************************/
/* ouvre un poste de travail                                         */
/*********************************************************************/
open_vwork()
{
int i;
        for(i=0;i<10;work_in[i++]=1);
        work_in[10]=2;
        handle=phys_handle;
        v_opnvwk(work_in,&handle,work_out);
}
```

```
/*************************************************************/
/* dess_loupe : dessin de la loupe                           */
/*************************************************************/

dess_loupe()
{int pxy[4];
        vsf_style(handle,8);
        vsf_interior(handle,1);
        vsf_color(handle,1);
        pxy[0]=97;
        pxy[1]=97;
        pxy[2]=223;
        pxy[3]=223;
        v_bar(handle,pxy);
}

raz_loupe()
{int pxy[4];
        vsf_style(handle,8);
        vsf_interior(handle,1);
        vsf_color(handle,0);
        pxy[0]=100;
        pxy[1]=100;
        pxy[2]=220;
        pxy[3]=220;
        v_bar(handle,pxy);

}

/*************************************************************/
/* Affiche_loupe : affichage de la loupe                     */
/*************************************************************/

Affiche_loupe(xm,ym) int xm,ym;
{register int xl,yl,i,j;
 int px[4];
 int pel,index;
        raz_loupe();
        vsf_style(handle,8);
        vsf_interior(handle,1);
        vsf_color(handle,1);
        yl=xl=101;
        for (i=ym;i!=ym+20;i++) {
                for (j=xm;j!=xm+20;j++) {
                        v_get_pixel(handle,j,i,&pel,&index);
                        if (index) {
                                px[0]=xl;
                                px[1]=yl;
                                px[2]=xl+4;
                                px[3]=yl+4;
                                v_bar(handle,px);
                                }
                        xl+=6;

                        }
                yl+=6;
                xl=101;
                }
}

/*************************************************************/
/*   boucle de la loupe, sortie sur mouse click              */
/*************************************************************/
```

```
loupe()
{int xa,ya;
 int xmouse,ymouse,pstatus,key;

        xa=ya=0;
        dess_loupe();
        while (1) {
                graf_mkstate(&xmouse,&ymouse,&pstatus,&key);
                if ((xa!=xmouse) || (ya!=ymouse))
                        Affiche_loupe(xmouse-25,ymouse-25);
                if (pstatus) break;
                xa=xmouse;ya=ymouse;
                }
}
/*************************************************************/
/*                    init de l accessoire                   */
/*************************************************************/
main()
{
        appl_init();
        phys_handle=graf_handle(&gl_wchar,&gl_hchar,&gl_wbox,&gl_hbox);
        menu_id=menu_register(gl_apid,"  Loupe  ");
        open_vwork();
        wind_get(0, WF_WORKXYWH, &xdesk, &ydesk, &wdesk, &hdesk);
        multi();
}

/*************************************************************/
/* taches de l accessoire                                    */
/*************************************************************/
multi()
{
int event;

      while (TRUE) {
        event = evnt_multi(MU_MESAG | MU_BUTTON | MU_KEYBD,
                           1,1,butdown,
                           0,0,0,0,0,
                           0,0,0,0,0,
                           msgbuff,0,0,&mx,&my,&ret,&ret,&keycode,&ret);


        if (event & MU_MESAG)
          switch (msgbuff[0]) {

          case AC_CLOSE:
                if (msgbuff[3]==menu_id)
                        v_clsvwk(handle);
                break;
          case AC_OPEN:
            if (msgbuff[4] == menu_id){
                open_vwork();
                graf_mouse(255,loupe_forme);
                loupe();
                graf_mouse(0,0L);
                v_clsvwk(handle);
              }
            break;

          } /* switch (msgbuff[0]) */

      } /* while (TRUE) */

}
```

ST MAG C

UN ZEST DE LOGO

LOGO est un langage intéressant né aux Etats-Unis au milieu des années 70 et qui souffre auprès des informaticiens de métier d'un vice rédhibitoire: il est facile!

C'est précisément pour sa facilité qu'il a été choisi dans certaines écoles pour apprendre aux enfants l'informatique - on devrait plutôt dire l'algorithmique. Logo dispose en effet de plusieurs atouts, dont l'un au moins est immédiatement disponible: son aisance graphique. C'est ce premier point que nous allons évoquer dans cet article d'initiation et nous réserverons pour la prochaine fois des subtilités plus professionnelles telles que le traitement des listes ou la récursivité.

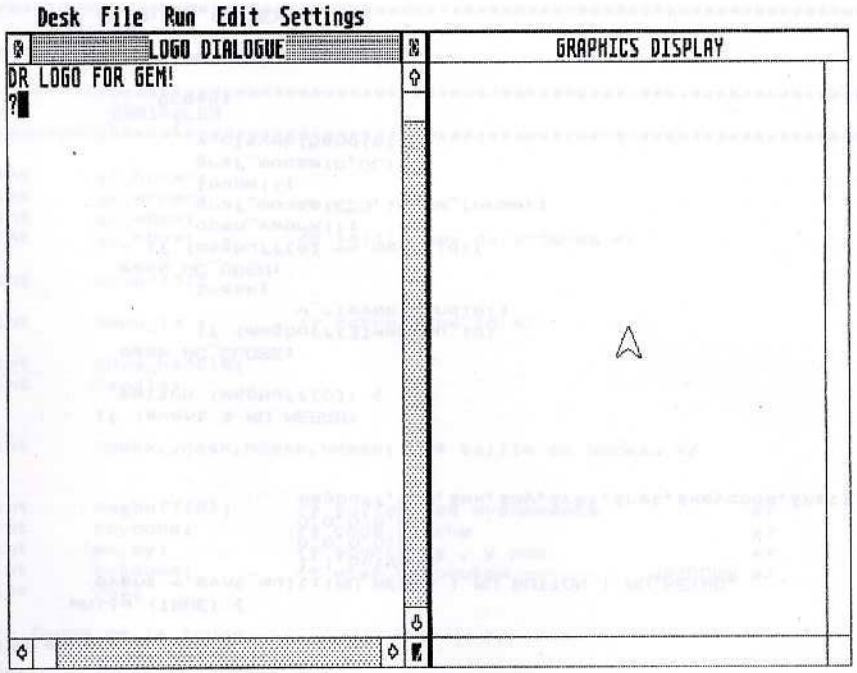
Il va de soi que l'aisance graphique ne pouvait laisser indifférent l'ATARI ST dont les multiples dons comportent entre autres le système GEM (Graphics Environment Management), autant dire qu'ils étaient faits l'un pour l'autre! Nous signalerons donc au passage les avantages particuliers du LOGO de l'ATARI ST.

L'ATARI ST hébergeait déjà une souris; vous avez sans doute pu voir chez les revendeurs ou vous verrez au SICOB que cette souris prend parfois son vol sous la forme d'une abeille (laborieuse, bien entendu), LOGO complète la ménagerie en le dotant d'une tortue! Cet animal sympathique, au demeurant fort stylisé à l'écran, et qui a pour nom anglais 'turtle', est en plus muni d'un stylo ('pen'). Il suffit de lui dire où aller et elle trace fidèlement un trait sur son passage.

Avant toute chose, installons-nous aux commandes, après avoir chargé LOGO.PRG. Voici l'aspect de l'écran (figure 1):

Le bureau, outre les commandes générales du haut de la page, comporte deux fenêtres actives: la fenêtre de dialogue et celle d'affichage graphique où attend la tortue, tête vers le haut de l'écran.

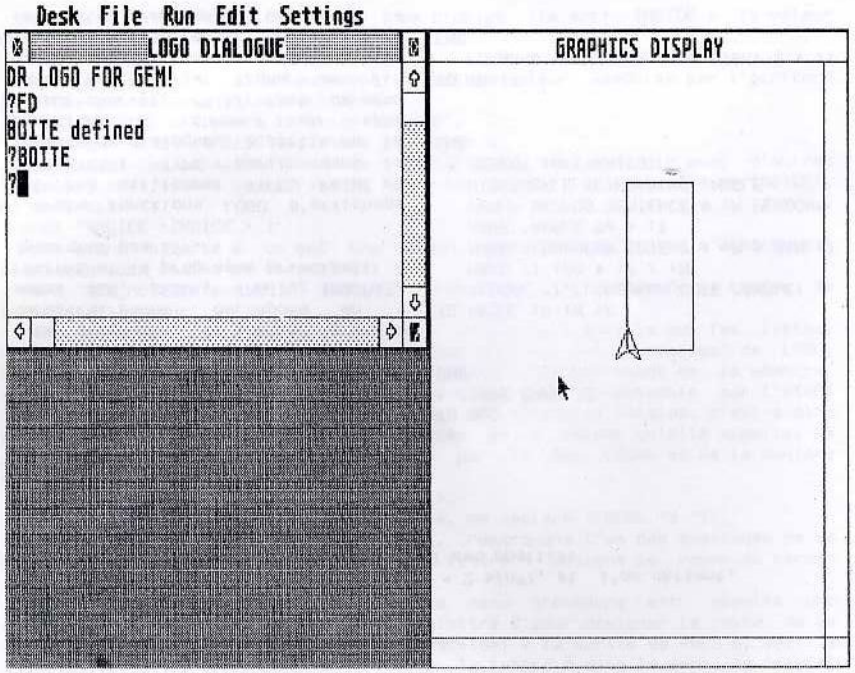
Il est possible d'écrire notre premier programme directement dans la zone de dialogue, mais comme les meilleur(e)s dactylographes commettent parfois des fautes de frappe et les meilleurs programmeurs des fautes de syntaxe, il est raisonnable de prendre l'habitude de rédiger les programmes au moyen de l'éditeur. Aucune difficulté avec le Dr LOGO d'ATARI puisque l'éditeur est présent en mémoire et répond au premier appel de la souris. Déposons sur la fenêtre de l'éditeur le résultat de nos fructueuses cogitations:




```
TO BOITE
REPEAT 2 [ FORWARD 100 RIGHT 90
           FORWARD 50 RIGHT 90 ]
END
```

Revenons en mode dialogue grâce à notre amie la souris, le mot 'BOITE' est maintenant défini (defined) et prêt à l'emploi. Employons-le donc en tapant au clavier BOITE <RETURN> et regardons la zone graphique (figure 2):

Analysons un peu le fonctionnement de ce mini-programme. LOGO, comme beaucoup de langages évolués tels que PASCAL, C, ou FORTH, permet à l'utilisateur de donner à ses procédures un nom de son choix. Alors qu'en BASIC, il faudrait identifier le sous-programme qui dessine un rectangle par l'appel de son numéro de ligne dans un GOSUB, ici l'appel de procédure a lieu dès l'emploi du mot BOITE, soit au clavier, soit, en différé, dans un programme. Cependant, LOGO, à l'inverse de PASCAL ou de C, n'est pas compilé mais interprété, ce qui rend ses performances plus proches de celles de BASIC en vitesse d'exécution, surtout pour les calculs. Heureusement, dans le cas des dessins réalisés par la tortue, l'usage par GEM et LOGO de routines très bien optimisées en langage machine et la vitesse du microprocesseur 16-32 bits donnent des résultats remarquables.



Reportons-nous à la création de la procédure: sa définition débute par le mot réservé TO suivi du nom choisi, et elle se termine par une ligne contenant le mot-clef END.

Dès que l'on vous aura précisé

que REPEAT 2 ordonne d'exécuter deux fois la liste des instructions qui suivent (entre crochets),

que FORWARD 100 demande à la tortue d'avancer de 100 unités graphiques élémentaires (pixels),

et que RIGHT 90 exige qu'elle tourne de 90 degrés vers la droite,

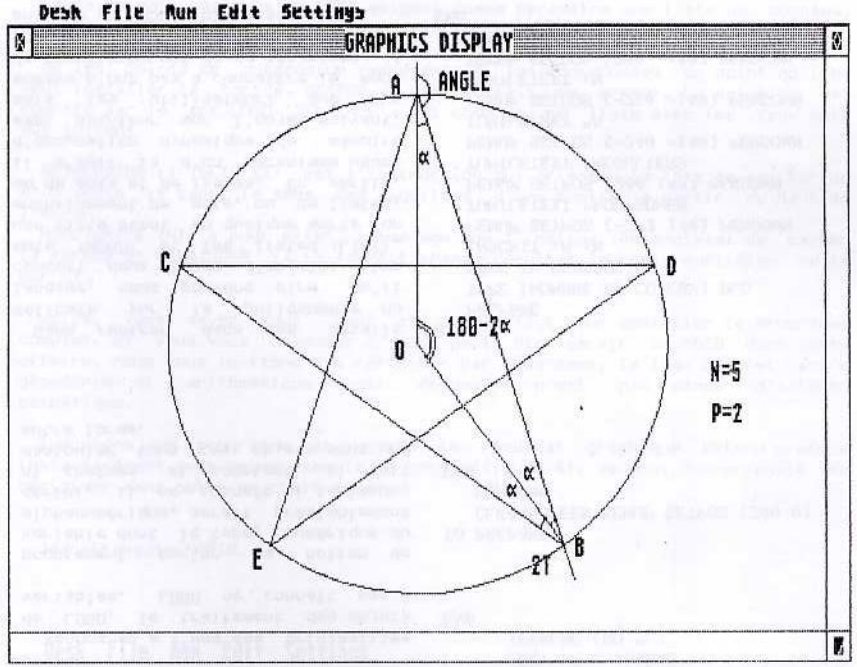
vous aurez complètement assimilé le sens du corps de la procédure.

Précisons pour les paresseux que LOGO admet pour beaucoup de mots réservés des abréviations bien commodes: ainsi FORWARD peut s'abréger en FD et RIGHT en RT. Toutefois, comme les mots-clefs complets sont généralement plus évocateurs, dans le programme qui suit, nous avons décidé de ne pas utiliser d'abréviations.

Maintenant que nous sommes plus familiarisés avec l'aspect d'une réalisation en LOGO, nous allons oser décortiquer ensemble un "vrai" programme.

Notre but est de faire dessiner sur l'écran noir et blanc de l'ATARI ST, car la résolution de l'image y est spectaculaire, tous les polygones réguliers à n cotés inscrits dans un même cercle de rayon R.

Décomposons la difficulté en étudiant le tracé du polygone régulier à n cotés obtenu en joignant les sommets possibles sur le cercle de p en p. La figure 3 illustrera notre propos (signalons au passage que la figure en question a été obtenue entièrement avec l'ATARI ST complété d'une imprimante (interface parallèle classique), y compris les adjonctions de texte sur le dessin).



Le pentagone étoilé ABCDE est obtenu en joignant les 5 points de 2 en 2, on obtiendrait le pentagone convexe en joignant les sommets de 1 en 1 par ADBEC.

Le cercle entier mesure 360°, nous ne préciserons plus que les mesures d'angle sont en degrés. L'angle $\widehat{AOB}$ mesure donc 360*p/n (ici les 2/5 de 360 soit 144).

Mettons-nous dans la peau (ou plutôt la carapace) de la tortue qui commence sa carrière, ne l'oublions pas, au centre de l'écran, soit en écran plein, à peu près en [200,0].

Emmenons-la en A et tournons-lui la tête pour qu'elle regarde B, sa prochaine destination; il nous faut calculer la mesure de l'angle ANGLE dont elle tourne vers la droite. Pour accomplir le dessin, il suffit, n fois de suite (ici 5), d'avancer en traçant un trait jusqu'au prochain sommet (la distance à parcourir AB=LONGUEUR_COTE est à calculer) et de tourner la tête de la tortue vers le sommet suivant (d'un angle dont la mesure 2T est à calculer).

Le lecteur rebuté par la géométrie classique peut omettre les explications des prochains paragraphes et se reporter aux résultats du calcul des valeurs nécessaires ANGLE, LONGUEUR_COTE et T.

Avec les notations de la figure, l'angle au centre $\widehat{AOB}$ mesure 180-2α (dans le triangle AOB), et aussi 360p/n, d'où

$$180-2\alpha = 360p/n$$
$$2\alpha = 180-360p/n$$

ou encore $\alpha = 90 - 180p/n$

d'autre part, l'angle plat de sommet B apparent sur la figure nous donne:

$$2T+2\alpha = 180$$

soit $\alpha = 90-T$

en rapprochant ces deux écritures, on a tout à la fois déterminé T et α:

$$T = 180\ p / n$$

et $\alpha = 90 - T.$

L'angle plat de sommet A nous permet de calculer ANGLE:

$$ANGLE = 180-\alpha$$
$$ANGLE = 180-(90-T)$$

d'où ANGLE = 90 + T.

Quant à la distance AB, elle résulte par exemple d'une formule trigonométrique dans le triangle, généralement connue sous la forme:

$$a^2=b^2+c^2-2bc.\cos(A),$$

ici, dans le triangle AOB,

$$AB^2=OA^2+OB^2-2\ OA.OB.\cos(180-2\alpha)$$

on sait que OA=OB=R et que $180-2\alpha=2T$, donc

$$AB^2=2\ R^2.(1-\cos 2T)$$

comme chacun sait (ou a su), $1-\cos 2T = 2\sin^2 T$, et l'on obtient, puisque $2T<180$, donc que $T<90$ et que sin T est positif:

LONGUEUR COTE = 2 R sin T.

Après l'effort, le réconfort, voici la programmation proprement dite:

Le programme ci-contre a été (arbitrairement) décomposé en 7 mots: BLC, REGUL, TRAIT, INSCRIT, PREPARE et le programme principal: Z (pourquoi pas Z?).

Trois fonctions LOGO d'affichage ont été utilisées: TYPE, PRINT et TURTLETEXT; TYPE écrit dans la fenêtre de dialogue, à partir de la position du curseur et sans retour-chariot, PRINT fait de même mais va à la ligne en fin d'exécution, quant à TURTLETEXT, cette fonction écrit du texte dans la fenêtre graphique, à partir de la position courante de la tortue - à la fin de l'écriture, la tortue n'a pas bougé -. Cette possibilité de mélanger texte et dessin est une intéressante particularité dont nous avons fait usage pour la figure 3.

Nous pouvons dès maintenant expliquer le premier mot: BLC affiche un espace à la position du curseur (les fanatiques du code ASCII avaient reconnu CHAR 32!).

Le mot PREPARE contient 4 fonctions LOGO simples:

CLEARSCREEN (CS en abrégé) qui efface l'écran graphique et remet la tortue au point de coordonnées (0,0), tête vers le haut.

PENUP (PU) qui lève le stylo, la tortue peut alors se déplacer sans écrire,

SETPOS [200,0] qui envoie la tortue au point de coordonnées (200,0), soit approximativement au centre de l'écran,

PENDOWN (PD) qui baisse le stylo, la tortue est alors prête à tracer.

Le sens de PREPARE est maintenant tout à fait évident.

Venons-en à l'une des originalités de LOGO, le traitement des objets variables. LOGO ne connaît pas à proprement parler la notion de variable dont le type, numérique ou alphanumérique, serait préalablement défini, il ne connaît ni tableaux, ni chaînes et pourtant il sait manipuler tous ces objets sous une autre forme.

Sans rentrer dans des détails délicats sur la philosophie du langage, nous pouvons dire qu'il conçoit deux types d'objets: les mots (WORD) et les listes (LIST), une liste étant en quelque sorte un enchaînement de mots, ou de listes, ou de mots et de listes. En vérité, il s'agit là d'un mécanisme usuel d'occupation dynamique de mémoire avec pointeur sur l'objet suivant, mais les utilisateurs que nous sommes n'ont pas à connaître le mode de fonctionnement de ce mécanisme (à l'inverse de l'usage des pointeurs en C ou en PASCAL par exemple).

```
TO BLC
   TYPE CHAR 32
END

TO REGUL :R :N :P
   (LOCAL "T "LONGUEUR_COTE "ANGLE)
   MAKE "T 180 * :P / :N
   MAKE "LONGUEUR_COTE 2 * :R * SIN :T
   MAKE "ANGLE 90 + :T
   PENUP SETPOS SENTENCE 0 :R PENDOWN
   HIDETURTLE SETHEADING :ANGLE
   REPEAT :N [TRAIT]
END

TO TRAIT
   FORWARD :LONGUEUR_COTE RIGHT 2 * :T
END

TO PGCD :A :B
   LOCAL "R
   LABEL "L
   TEST (:B=0)
   IFTRUE  [OUTPUT :A]
   IFFALSE [MAKE "R
            (REMAINDER :A :B)
            MAKE "A :B
            MAKE "B :R
            GO "L]
END

TO INSCRIT :R :N
   PRINT [CE PROGRAMME DESSINE LES
   POLYGONES]
   TYPE [REGULIERS A] BLC TYPE :N
   PRINT [COTES INSCRITS]
   TYPE [DANS UN CERCLE DE RAYON]
   BLC TYPE :R PRINT ".
   CLEARSCREEN
   LOCAL "BORNE MAKE "BORNE :N / 2
   LOCAL "P MAKE "P 1
   LABEL "L
      TEST (PGCD :N :P) = 1
     IFTRUE  [REGUL :R :N :P]
      IFFALSE []
      MAKE "P :P + 1
      TEST :P > :BORNE
      IFFALSE [GO "L]
END

TO PREPARE
   CLEARSCREEN PENUP SETPOS [200 0]
   PENDOWN
END

TO Z
   PREPARE
   TYPE [NOMBRE DE COTES?] BLC
   MAKE "N READQUOTE
   INSCRIT :R :N
   PENUP SETPOS [-272 140] PENDOWN
   TURTLETEXT "POLYGONES
   PENUP SETPOS [200 140] PENDOWN
   TURTLETEXT "REGULIERS
   PENUP SETPOS [-240 -140] PENDOWN
   TURTLETEXT "A
   PENUP SETPOS [-216 -140] PENDOWN
   TURTLETEXT :N
   PENUP SETPOS [200 -140] PENDOWN
   TURTLETEXT "COTES
END
```

Ainsi, si l'on veut associer le nom d'objet (le mot) INDICE à la valeur numérique 0, on écrira:

MAKE "INDICE 0

et l'on accèdera ultérieurement à cette valeur associée par l'écriture :INDICE, par exemple l'instruction

PRINT "INDICE affichera le texte 'INDICE',

alors que PRINT :INDICE affichera la valeur 0.

Pour augmenter la valeur associée à INDICE de 1, (par analogie avec d'autres langages, nous dirons improprement: pour incrémenter le 'contenu' de INDICE), on écrirait:

MAKE "INDICE :INDICE + 1

Nous avons affecté à un mot une valeur numérique, mais rien n'empêcherait d'affecter une chaîne de caractères:

MAKE "MON_MOT "auteur PRINT :MON_MOT produirait l'affichage de la chaîne de caractères 'auteur'.

Nous donnerons plus d'explications dans un prochain article sur les listes, nos connaissances sont déjà suffisantes pour utiliser les "variables" de LOGO. Une autre précision sur ce sujet: afin d'éviter l'encombrement de la mémoire, ce qui est une bonne habitude, même si ce n'est pas indispensable sur l'ATARI 520 ST, il est souhaitable d'utiliser des variables locales, c'est-à-dire propres à la procédure qui les utilise et à celles qu'elle appelle. La déclaration s'en fait avant l'emploi, par le mot LOCAL et de la manière suivante:

pour un seul nom "X, on déclare LOCAL "X,

pour plusieurs noms "X et "Y par exemple, on déclare (LOCAL "X "Y).

Sans vouloir insister sur ce sujet, remarquons l'un des avantages de ce procédé: dans la procédure INSCRIT, la lettre R désigne le rayon du cercle circonscrit aux polygones; dans cette même procédure est appelée une sous-procédure PGCD qui utilise la même lettre R pour désigner le reste de la division de A par B dans l'algorithme d'Euclide; à la sortie de PGCD et dès la ligne suivante, le programme retouve bien la lettre R dans le sens de 'rayon' et avec la bonne valeur. A titre de curiosité, offrez-vous la même fantaisie en BASIC!

Une autre commodité de LOGO est le passage de paramètres depuis une procédure appelante à une procédure appelée: ainsi PGCD a pour en-tête

TO PGCD :A :B

où :A et :B sont des valeurs dont on veut calculer le PGCD; pour calculer le PGCD de 1230 et de 450, il suffira d'appeler PGCD 1230 450, ou encore, si l'on a précédemment affecté ces valeurs à des mots par MAKE "N1 1230 MAKE "N2 450, on calculera leur PGCD en appelant PGCD :N1 :N2. A et B sont ainsi des variables muettes de la "fonction" PGCD.

Pour que notre compréhension du programme soit complète, il nous reste à dire quelques mots sur les branchements conditionnels, sur les entrées de données au clavier, sur quelques fonctions graphiques et algébriques non encore définies et sur la façon de former une liste avec deux mots.

LOGO n'admet ni boucle POUR (FOR), ni instruction TANT QUE (WHILE), ni JUSQU'A (UNTIL) mais toutes ces instructions de structuration, fortement recommandées en programmation, peuvent être simulées à l'aide de TEST... IFTRUE...IFFALSE...et de GO "étiquette. Nous avons utilisé cette construction dans PGCD et dans INSCRIT.

Les entrées de données au clavier peuvent se faire en LOGO au moyen de trois mots-clefs: READCHAR (RC), READQUOTE (RQ), et READLIST (RL) qui lisent au clavier respectivement un caractère, un mot et une liste; dans la procédure Z, nous avons introduit la valeur du nombre N de côtés par les instructions:

MAKE "N READQUOTE .

Certains mots réservés de LOGO exigent comme paramètre une liste de données, c'est le cas de SETPOS dont la syntaxe d'emploi est:

SETPOS [... , ...]

où les points de suspension représentent les coordonnées du point où l'on veut installer la tortue. SETPOS 0 0 produirait une erreur. La solution est fournie par le mot LOGO 'SENTENCE' qui compose une liste avec les deux mots qui suivent.

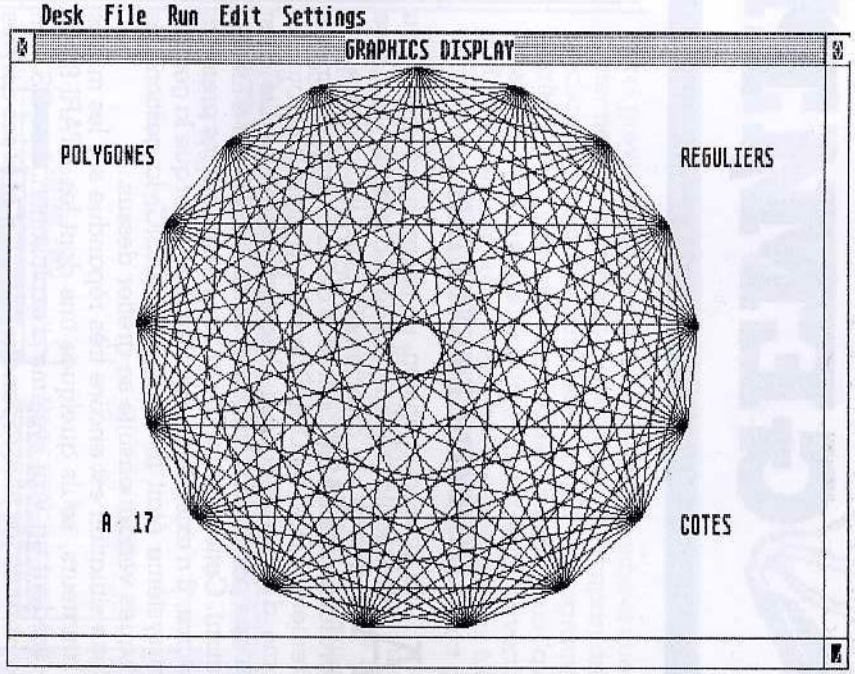
SETHEADING :X (SETH :X) est l'instruction qui oriente la tête de la tortue d'un angle :X dans le sens des aiguilles d'une montre à partir du haut de l'écran.

HIDETURTLE (HT) a pour effet, comme son nom l'indique (en anglais) de cacher la tortue et REMAINDER :X :Y fournit comme résultat le reste euclidien de :X par :Y.

Voilà, vous savez tout! tout ce qu'il faut pour assimiler le programme complet. Si vous vous demandez à quoi peut bien servir un PGCD dans cette affaire, nous vous invitons à y réfléchir par vous-même, le lien naturel entre géométrie et arithmétique ainsi découvert n'est pas dénué d'intérêt esthétique.

Exécutons ce programme et voyons le résultat graphique obtenu presque instantanément avec comme exemple 17 côtés (figure 4): de quoi donner envie de récidiver dans notre prochain numéro.

Jean-François RAPIN.



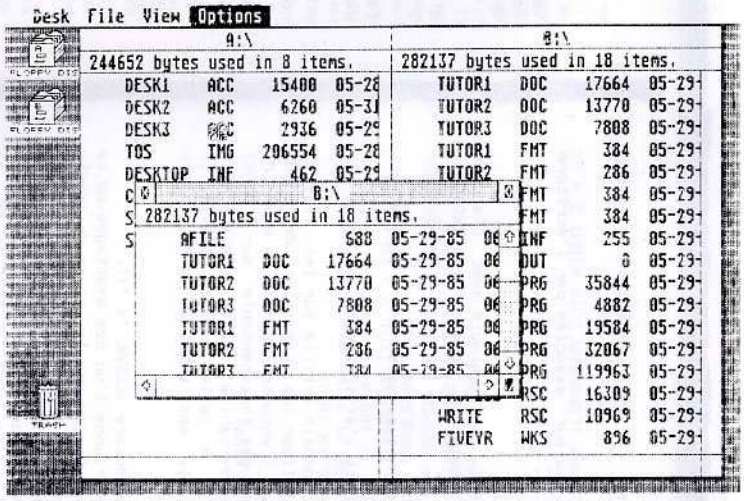
GEM, POURQUOI

1 - AU COMMENCEMENT ETAIT...

A la mise sous tension, un micro se présente à peu de détails près suivant deux cas de figure. On se retrouve soit directement dans un langage, généralement le BASIC, soit sous ce qu'on appelle improprement le DOS ou Disk Operating System (SED ou Système d'Exploitation des Disques) qui en réalité n'est qu'une partie de l'Operating System (Système d'Exploitation). Cette confusion vient de ce que sur les premiers micros, il n'existait pas de véritable OS et que la gestion du système était mélangée au BASIC, la gestion des disques venant ensuite se greffer dessus.
Cette situation est encore très répandue sur les micros d'amateurs, seuls quelques uns dont les ATARI 8 bits possèdent un vrai système d'exploitation. Les OS sont nombreux, les plus répandus étant CP-M et ses dérivés sur les 8 bits, MS-DOS et CP-M sur les 16 bits. Il en existe d'autres telsque UNIX et OASIS qui sont des produits plus élaborés.
Tout ces logiciels se présentent de la même façon au regard de l'utilisateur : un message de sollicitation souvent très court qui informe qu'une ligne de commande est attendue. Celle-ci est généralement formée d'un nom de commande suivie de paramètres comprenant des noms de fichiers et des options. L'introduction de la commande entraîne l'exécution d'un programme déjà présent en mémoire ou chargé depuis un disque. Ce programme pouvant être aussi bien celui donnant la liste des fichiers présents sur un disque qu'un langage de programmation ou un traitement de texte. Il faut noter que le programme qui accepte la ligne de commande est déjà en lui-même une commande ou quelque chose de similaire.
Sur des systèmes simples la liste des commandes est restreinte et peut être facilement mémorisée, la syntaxe du passage des paramètres aussi. Sur des systèmes plus évolués le nombre de commandes est nettement plus impressionnant (un recensement rapide sur un système que j'utilise quotidiennement donne 80 commandes et 377 options sans compter les variantes de syntaxe).
Afin de permettre au programmeur de s'y retrouver, les concepteurs de systèmes ont choisi en gros, soit d'inclure une fonction guide qui rappelle l'usage de la commande, sa syntaxe et ses options, soit de poser successivement les questions amenant l'utilisateur à préciser ce qu'il désire. Ces deux principes ont leurs avantages et leurs inconvénients. Le premier permet de ne pas perdre de temps quand on sait ce que l'on veut, avec l'inconvénient de pouvoir lancer une commande en n'étant pas certain de son action, le deuxième s'il présente plus de garanties de sécurité est lourd et fait perdre beaucoup de temps à ceux qui savent ce qu'ils veulent.

2 - L'AVENEMENT DU GRAPHISME

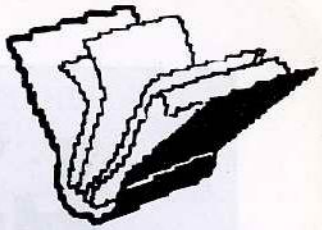
Avec l'augmentation de capacité graphique des micros un autre mode est apparu. Partant du principe qu'un micro utilisé en gestion s'occupe principalement de dossiers que l'on classe, consulte, détruit... Rank Xerox a construit une machine reproduisant l'environnement de l'employé de bureau. Des bacs à fiches, une corbeille et des feuilles de papiers que l'on dispose à sa convenance avec, révolution, non pas les touches du clavier, mais un prolongement de la main se déplaçant avec la même aisance que la main elle-même : la souris. Tout les ingrédients de la réussite y étaient sauf le prix. C'est Apple et Microsoft avec le Mac Intosh qui tireront les prix vers le bas avec le succès que l'on sait. Digital Research a rapidement répliqué avec GEM pour adapter le même principe aux autres systèmes (IBM et compatibles) en utilisant le savoir faire qu'elle avait montré avec CP-M et son logiciel graphique GSX. Avec un plus : la conformité des commandes graphiques avec la norme ANSI.



3 - GEM

GEM, comme l'indique son nom complet, Graphics Environment Manager, n'est pas un système d'exploitation, mais un gestionnaire d'environnement graphique. Il vient s'intercaler entre l'OS et l'utilisateur en remplaçant en quelque sorte le programme d'exécution de commandes. Mais ceci n'est que la partie émergée, dans un autre article nous plongerons à l'intérieur.
Comment se présente donc le bureau ? Tout en haut quelques mots « Bureau Fichier Options Spécial ». Si la petite flèche qui symbolise la main sur le bureau touche un de ces mots, un menu apparaît, différent pour chacun des mots. Certaines des lignes du menu sont écrites en gris et sont inactives, en revanche, celles écrites en noir sur fond blanc sont inversées par un contact avec la souris. En appuyant sur le bouton (cliquant, néologisme) le menu s'efface et l'option choisie dans le menu est immédiatement activée. Certaines options sont précédées d'un marqueur indiquant qu'elles sont actives,

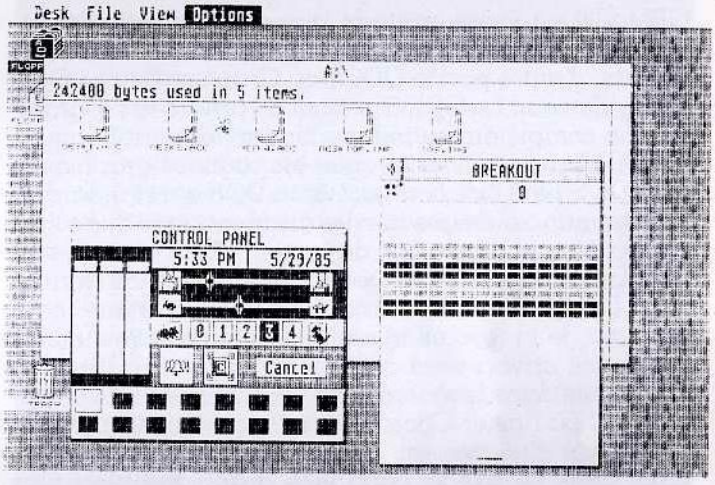
ET COMMENT



comme par exemple celle indiquant l'ordre de tri de la liste des fichiers. Pour voir cette liste pas d'ordre à taper. Chaque lecteur de disque est représenté par un tiroir contenant des chemises, celles-ci représentant les fichiers. Les chemises peuvent être regroupées en dossiers qui sont des sous-catalogues. Tout comme on utilise des chemises de couleurs différentes, les symboles sont différents suivant le type du fichier (programme ou données). Pour voir le contenu d'un tiroir il faut l'ouvrir en sélectionnant d'abord le tiroir (clic), en touchant le mot « fichier » (le menu apparaît) puis le mot « ouvrir » (reclic) et... une feuille semble sortir du classeur et envahir l'écran. La liste des fichiers se trouve sur cette feuille appelée aussi fenêtre. Trop long tout ça ? Alors plus simple, toucher le tiroir, clic-clic et vous obtenez le même résultat. Le double clic indique à GEM que vous souhaitez activer un processus, dans ce cas la visualisation du catalogue du disque. Il est possible d'ouvrir un autre tiroir sans refermer le précédent, une deuxième fenêtre venant s'afficher à côté et même un peu au-dessus de la première. Si vous désirez revoir celle-ci au dessus de la pile il suffit de la toucher et de cliquer une fois. Vous pouvez la pousser sur le côté comme vous mettez un document sur un coin de bureau, la plier pour qu'elle prenne moins de place (ici vous réglez la taille que vous voulez) ou la remettre dans son tiroir qui sera s'il le faut refermé.

Et si je veux exécuter un programme ? Toucher l'icône ou le nom du programme, clic-clic, et le programme envahit l'écran comme l'avait fait la liste tout à l'heure. Si ce programme est écrit avec la méthode GEM vous ne serez pas dépaysé, vous vous retrouvez avec un bureau organisé différemment, comprenant une nouvelle barre de menu et d'autres icônes. Par exemple sous Logo pour charger un programme, toucher « fichier », cliquer sur ouvrir, la liste des programmes en BASIC disponibles apparaît dans une boîte, pour charger l'un d'eux, double-cliquer dessus et c'est tout. Si au contraire vous désirez en sauvez un, toucher « fichier », cliquer sur « sauver », la liste des programmes apparaît aussi, vous pouvez désigner un programme existant ou taper un nouveau nom. Pour lister un fichier de données sur imprimante, double-cliquer sur le nom, une boîte apparaît et prévient que ce n'est pas un programme et que ce fichier ne peut qu'être listé sur écran ou sur imprimante. Un clic et c'est parti. Quand l'exécution d'un programme s'arrête, vous retrouvez votre bureau tel que vous l'avez laissé. Le gestionnaire de bureau autorisant l'ouverture de 7 fenêtres, il en sera probablement des bureaux électroniques comme des vrais, ordonnés ou fouillis au gré de l'utilisateur.

Pour effacer un fichier on le jette dans la corbeille, mais attention, inutile d'essayer d'aller y fouiller, cette corbeille gloutonne digère tout. Ce qui est mis dedans est irrécupérable.

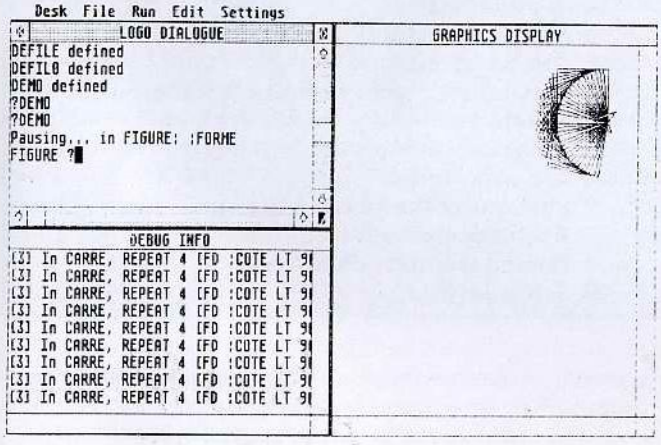


4 - LE COTE PRATIQUE

Et la rapidité ? Elle est fonction de celle de la machine et sur le 520 ST, ça décoiffe. Pour vous donner une idée, les menus sont dits déroulants et en effet sur certaines machines ou les voit se dérouler du haut vers le bas en raison de temps nécessaire au transfert de mémoire écran, sur le ST non. J'ai beau essayer de bien faire attention, rien à faire les menus semblent apparaître d'un bloc. De même pour les boîtes de dialogue ou d'alerte qui surgissent du néant et y retournent aussi vite. Quant à la rapidité de manipulation, c'est une question d'habitude. Le plus difficile est d'éviter de toucher les mots de la barre de menu, car il faut ensuite s'éloigner et cliquer sur le bureau pour faire disparaître le menu, sinon le reste ne demandant qu'un positionnement approximatif ne pose aucun problème.

Est-ce réellement plus pratique ? Il est certain qu'un programmeur expérimenté va moins vite avec un logiciel de ce type pour les commandes auquel il est habitué, mais ceci est rapidement compensé par le gain de temps réalisé sur toutes les fonctions spéciales ainsi que par la plus grande sécurité apportée par l'affichage des paramètres. Pour l'utilisateur et si le programmeur a conçu son programme dans l'optique GEM, à part la commande vocale, je ne vois rien de mieux pour l'instant.

Jean-Michel DUBOIS



LE GRAPHISME DE GEM

Le logiciel GEM inclus dans le système d'exploitation du 520ST peut être scindé en plusieurs parties. La gestion du graphique de base est confié au VDI ou Virtual Device Interface, qui est l'objet de cet article.

1 - ORGANISATION DE GEM

GEM VDI est divisé en deux, d'une part le code dépendant des périphériques graphiques utilisés ou Device Drivers, d'autre part le GDOS ou Graphics Device Operating System. Le GDOS permet de rendre la programmation complètement indépendante des périphériques utilisés. L'application invoque les routines graphiques de GDOS peut être comparé à un DOS qui standardise les commandes de gestion des disques quelle que soit la capacité ou la dimension de ceux-ci. Ainsi par son système de coordonnées virtuelles appeléé NDC ou Normalized Device Coordinates, il rend le programme indépendant de la résolution réelle du périphérique utilisé, les device drivers se chargeant de calculer l'adressage des points dans le système de coordonnées réelles appelé RC ou Raster Coordinates. Il est bien sûr possible d'adresser directement par les coordonnées réelles, mais dans ce cas la portabilité graphique n'est plus assurée, l'image pouvant subir des déformations ou des dépassements de résolution. Le système NDC à son origine dans le coin inférieur gauche du périphérique.
Les device drivers quant à eux sont en quelque sorte le BIOS (Basic In Out System) du graphisme, ils sont indissociables du périphérique qu'ils gèrent et sont chargés en mémoire par le GDOS en fonction des besoins. Ils assurent la traduction des ordres standards en ordres particuliers au périphérique ou la simulation des fonctions non supportées. Par exemple si une table traçante ne sait pas tracer un cercle, le device driver tracera le cercle comme une suite de vecteurs, sinon il enverra l'ordre : tracer un cercle de tel rayon à tel endroit. Le programmeur ou l'utilisateur ne fera pas la différence entre les deux modes, il ne connait que l'ordre standard de VDI qui en C s'écrit : v-circle(device,x,y,rayon).
Les ordres de tracés de graphisme peuvent aussi être stockés dans les fichiers et ensuite envoyés à un périphérique, ce type de fichier est appelé Metafile. Ils ne contiennent pas l'image mais le descriptif des éléments de l'image, forme, taille, couleur, remplissage etc. Les Metafile utilisent un périphérique idéal dont la résolution serait la même que celle du NDC, soit 32767 × 32767 points.
Les périphériques supportés peuvent être des écrans, des imprimantes, des tables traçantes ou des tablettes graphiques, appareils de reproduction photographique etc. Pour implanter un nouveau type de périphérique il suffit d'écrire le device driver adéquat.

2 - LES FONCTIONS DU GDOS

Les tableaux suivants vous donnent une idée de la puissance graphique de GEM, ils contiennent les fonctions et sous-fonctions de GEM, le chiffre indiquant le code de la fonction. La documentation de VDI fait exactement 300 pages, et n'oublions pas que ceci ne concerne que la gestion du graphique de base. La gestion des fenêtres, de la souris et des menus vient ensuite.

FONCTIONS DE CONTROLE	
1	Ouvrir la station de travail
2	Fermer la station de travail
3	Effacer la station de travail
4	Mettre à jour la station de travail
100	Ouvrir une station de travail virtuelle
101	Fermer une station de travail virtuelle
119	Charger un jeu de caractères
120	Décharger un jeu de caractères
129	Coordonnées des rectangles de clipping (zones de mise à jour)

FONCTIONS DE SORTIES	
6	Polyline (ligne brisée)
7	Polymarker (sommets de lignes brisées sans les lignes)
8	Texte
9	Remplissage de formes (même auto-sécante)
10	Tableau de pixels
103	Remplissage extérieur
114	Remplissage de rectangle
11	Primitives de dessin généralisées
	Sous-fonctions
1	Barre (rectangle)
2	Arc de cercle
3	Portion de cercle remplie (camembert)
4	Cercle
5	Ellipse
6	Arc elliptique
7	Portion d'ellipse remplie
8	Rectangle arrondi
9	Rectangle arrondi rempli
10	Texte justifié

Jean-Michel DUBOIS

FONCTIONS DE POSITIONNEMENT D'ATTRIBUTS

32 Mode d'écriture (transparent, opaque, ou exclusif, transparent inverse)
14 Réglage couleur
15 Type de lignes (continu, tireté long et court, trait d'axe, etc)
113 Définition d'un type de ligne
16 Epaisseur des lignes
17 Couleur des lignes
108 Terminaisons des lignes (carré, demi-cercle, flèches)
18 Type de polymarker (. + × etc)
19 Taille des polymarkers
20 Couleur des polymarkers
12 Hauteur des caractères
107 Hauteur des caractères en résolution d'imprimante (1/72 pouce)
13 Rotation des axes d'écriture de texte
21 Style des caractères
22 Couleur des caractères
106 Attributs spéciaux des caractères (gras, léger, souligné, liseré, etc)
39 Alignement des caractères
23 Style de trame de remplissage
24 Index de trame de remplissage
25 Couleur de trame de remplissage
104 Visibilité ou invisibilité du périmètre de remplissage
112 Trame définie par l'utilisateur

FONCTIONS RASTER

109 Copie de bloc opaque
121 Copie de bloc transparent, ou exclusif, transparent inverse
110 Passage de NDC en RC et vice-versa
105 Valeur d'un pixel

FONCTIONS D'ENTREE

33 Mode d'entrée (demande ou saisie au vol)
28 Demande de positionnement du curseur
Idem par saisie au vol
29 Demande de positionnement d'un nombre dans un intervalle
Idem par saisie au vol
30 Demande de choix
Idem par saisie au vol
31 Entrée d'une chaîne de caractères
Idem par saisie au vol
111 Définition de la forme de la souris
118 Permute le vecteur d'interruption d'un timer
122 Montre le curseur
123 Cache le curseur
124 Etat des boutons de la souris
125 Permute le vecteur de changement d'état des boutons
126 Permute le vecteur des déplacements de la souris
127 Permute le vecteur du déplacement du curseur
128 Etat des touches shift de droite et de gauche, control et alternate

FONCTIONS DE DEMANDE D'ETAT D'ATTRIBUT

102 Liste des informations spécifiques à un périphériques
26 Réglage couleur
35 Attributs des lignes
36 Attributs des polymarkers
37 Attributs du remplissage
38 Attributs du texte
116 Encombrement d'un texte donné
117 Taille des caractères
130 Style des caractères
27 Tableau de pixels
115 Mode d'entrée
131 Attributs spéciaux des caractères

FONCTIONS SPECIALES (ESCAPE)

Ces sous-fonctions sont spécifiques à un périphérique.

Pour les écrans
1 Demande le nombre de lignes et de colonnes en mode texte d'un écran
2 Sortie du mode texte
3 Passage en mode texte
4 Passage à la ligne précédente du curseur en mode texte
5 Idem ligne suivante
6 Idem colonne suivante
7 Idem colonne précédente
8 Curseur en haut à gauche
9 Effacement depuis le curseur à la fin de l'écran
10 Effacement depuis le curseur à la fin de la ligne
11 Positionnement du curseur
12 Affichage de texte
13 Vidéo inverse
14 Vidéo normale
15 Demande la position du curseur
18 Positionnement du curseur graphique
19 Suppression du curseur graphique

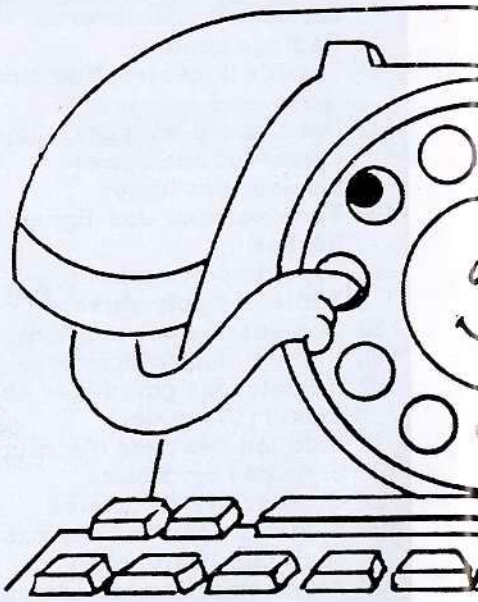
Pour les tablettes, joysticks, souris etc
16 Demande l'état d'une tablette...

Pour les imprimantes
17 Copie papier
20 Saut de page
21 Imprime une zone de l'écran
22 Annule les impressions en attente
23 Imprime la copie de mémoire d'écran contenue dans un fichier

Pour la recopie photographique Polaroïd palette
91 Demande le type de film
92 Demande l'état du driver
93 Change l'état du driver
94 Sauve l'état du driver dans un fichier
95 Supprime les messages d'information
96 Affiche un message d'information

Pour les Metafiles
98 Modifie l'indication d'encombrement d'une image en Metafile
99 Ecrit un enregistrement défini par l'utilisateur en Metafile
100 Change le nom d'un Metafile

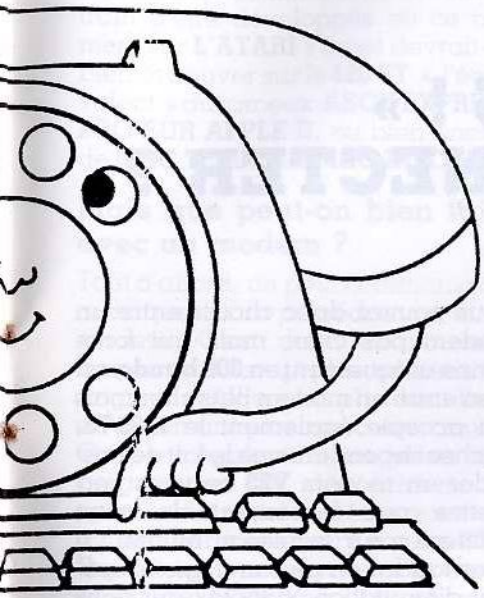
COMMUNICATION THEORIE...



Si l'habitude de communiquer entre micro-ordinateurs ou entre un micro et un ordinateur est peu répandue, c'est surtout en raison de l'absence de l'interface RS 232 en standard et surtout du prix élevé des modems en France. Ces derniers sont généralement des appareils professionnels, avec des performances élevées, de nombreuses possibilités de configuration, et un marché restreint. L'amateur lui se satiferait volontiers d'appareils plus rudimentaires et moins coûteux. L'accroissement du parc de micros équipés d'interface série devrait amener une modification de cette tendance. Il reste un problème à résoudre : quel modem choisir ? La question qui se pose en sous-entends immédiatement d'autres : pour quoi faire ? Comment ça marche ?
Il faut du matériel et du logiciel.

1 - COMMENT ÇA MARCHE

Un modem sert à adapter les signaux « tout ou rien » des ordinateurs en signaux compatibles avec une transmission sur une longue distance. De plus dans le cas qui nous intéresse, ces signaux doivent être véhiculés par le réseau téléphonique appelé réseau commuté par opposition aux réseaux spécialisés dans les transmissions de données (**TRANSPAC, CADUCEE**,...). Le réseau commuté étant conçu pour transmettre la parole, ses circuits sont calculés pour ne laisser passer que les fréquences utiles à la compréhension, soit **300 à 3400 hz**. Dans cette plage de fréquence réduite il est difficile de passer beaucoup d'informations par unité de temps. Le standard de modem le plus répandu à l'heure actuelle est appelé **300-300**, tout simplement parce qu'il est capable de transmettre **300 informations élémentaires (0 ou 1)** par seconde dans chaque sens, cequi donne en gros trente caractères par seconde. Cette faculté de communiquer simultanément dans les deux sens et de s'y retrouver est nommée duplex intégral (en anglais **full-duplex**) par opposition au mode où un seul à la fois des deux modems est autorisé à transmettre ou semi-duplex (**half-duplex**). Ce mode n'et pour ainsi dire pas utilisé sur les micros car peu pratique. La confusion est souvent faite entre mode full/half duplex et le renvoi ou non de l'écho (affichage à l'écran) des caractères tapés au clavier. Cela n'a pourtant rien à voir. La confusion vient du fait qu'en général les serveurs en duplex intégral renvoient l'écho avec terminaux alors que les serveurs en semi-duplex ne le renvoient pas.
Il arrive quelque fois qu'un serveur en duplex intégral ne renvoie pas l'écho, en revanche le renvoi d'écho par un serveur en semi-duplex s'il est possible donnerait des temps de transmissions catastrophiques. De plus, la notion d'écho n'a de sens que si un terminal est raccordé au modem.
Si la vitesse de trente caractères/seconde paraît rapide à côté de celle de nos doigts sur le clavier, elle est tout à fait insuffisante pour de nombreuses applications. La course à la vitesse a été engagée, **600-600, 1200-1200, 2400-2400** et maintenant **4800-4800** sont des vitesses exploitables par téléphone. Mais le prix a suivi. Un modem **4800-4800** sur réseau commuté est une petite merveille d'électronique et de micro-informatique coutant environ **35000 francs**. Une autre solution consiste à utiliser une vitesse différente à l'émission et à la réception. C'est celle choisie par le **Vidéotex**, la vitesse du serveur vers le **Minitel** est de **1200 bits/seconde** et du **Minitel** vers le serveur de **75 bits/seconde**. Certains Minitels et serveurs peuvent être « retournés », et les vitesses sont permutées. Ces modes sont des faux duplex intégral, et les spécialistes les appellent semi-duplex avec voie de retour ou voie de service. Ils posent un problème supplémentaire dans la mesure où rares sont les ordinateurs acceptant de fonctionner à des vitesses différentes à l'émission ou la réception. Il ne s'agit pourtant pas d'une impossibilité technique, la majortié des circuits intégrés d'interface ayant cette faculté, mais de l'inexploitation de cette possibilité par les constructeurs de micros. Ceci permet d'ailleurs aux fabricants de modems de proposer des appareils dits symétrisés qui font eux-mêmes la transformation **1200-75** ou **75-1200** en **1200-1200**, et cela se paie.
Tous ces modems même de marques différentes, peuvent communiquer sans problèmes avec un modèle équivlent, grâce au respect des normes internationales diffusées par le **CCITT (Comité Consultatif International du Téléphone et du Télégraphe)**, sauf si un pays décide de ne pas respecter les avis d'un organisme dont il fait pourtant partie, car ce ne sont que des avis. Et le problème c'est qu'il s'agit des **USA**. Donc les **modems US** s'ils peuvent parfaitement fonctionner entre eux en France, ou appeler une banque de données aux **USA** (attention à la prochaine facture de téléphone) sont incapables d'entrer en liaison avec des modèles **CCITT** bien que leurs vitesses soient les mêmes. Les fréquences sont différentes et ils se révèlent mutuellement sourds. Afin d'éviter tout malentendu sachez simplement que les normes **CCITT sont identifiées par un « V »** suivi du numéro d'avis **(V21=300−300, V23=1200−1200 ou 1200−75 etc)** et les normes **US, « BELL »** suivi d'un numéro **(BELL 103, BELL 202, etc).**



2 - COMMENT TRANSMET T'ON

Les octets qui sont maintenant les mots ou fractions de mots les plus utilisés en micro-informatique sont envoyés bit par bit sur la ligne téléphonique. La transformation du mot en bits séparés est effectuée par l'interface série qui modifie en outre les tensions utilisées. Le modem transforme ces signaux numériques en variations de fréquence, d'amplitude ou de phase de signaux sinusoïdaux que le réseau commuté transmet. A l'autre bout de la chaîne qui est symétrique, la transformation inverse est effectuée. Il est nécessaire que les équipements soient synchronisés afin de se comprendre ; deux modes sont utilisés. En synchrone les caractères à transmettre sont regroupés en blocs auxquels sont ajoutés des caractères de synchro et des délimiteurs de blocs. Ce mode n'est utilisé qu'à partir de **1200 bits/seconde.** En asynchrone chaque caractère comporte deux ou trois bits supplémentaires, un bit de start et un ou deux bits de stop. Peu de micro travaillant en synchrone, certains modems effectuent eux-mêmes la transformation asynchrone/synchrone et vice versa.

Quel que soit le mode, il est possible d'adjoindre à chaque caractère une information permettant de tester son intégrité après transmission **(VRCC)**, la solution choisie est de compter si le nombre de bits à « un » d'un mot est pair ou impair, puis de l'exprimer par un bit supplémentaire. On peut décider de noter par « un » la parité ou l'imparité du mot, ce qui donne deux codages possibles, la parité paire **(Even)** ou impaire **(Odd)**.

3 - QUE PEUT-ON FAIRE AVEC UN MODEM

L'utilisation d'un modem avec un micro-ordinateur, peut se résumer à trois cas.

— La simulation de terminal informatique pour pouvoir consulter les informations d'un autre ordinateur, ce dernier pouvant ou non utiliser le format Vidéotex. Ce mode ne permet pas de conserver la trace des échanges effectués avec le serveur. Un émulteur **VT-52**, terminal répandu, existe sur les **ST.**

— La simultion avec possibilité de recopie sur disque ou d'envoi de fichier disque s'apparente à l'utilisation précédente, mais en plus les échanges peuvent être consultés ultérieurement sans communication téléphonique.

Ces deux modes sont juste suffisants pour des consultations visuelles, car toute erreur de transmission due à des bruits parasites sur les lignes se traduira soit par la détection par le matériel ou le logiciel d'une erreur de parité soit par un caractère erroné car le test de parité très simple est aussi d'une fiabilité faible.

— La transmission de fichiers fait appel à des techniques plus évoluées de contrôle de transmission. Les données n'étant pas à transmettre caractère par caractère mais par enregistrement, il est possible d'utiliser des algorithmes de vérification nettement plus élaborés. Au contrôle de parité par mot **(VRCC)**, on peut ajouter un calcul de parité sur l'ensemble des mots d'un bloc **(LRCC)** ou même représenter le message par un polynôme fonction de la position des bits à « un ». Il existe d'autres codes encore plus complexes qui vont jusqu'à transmettre le bloc avec une redondance partielle permettant de détecter avec précision l'endroit de l'erreur et même quelquefois de reconstituer le mot initial. En général, les logiciels courants se contentent du **VRCC** et du **LRCC** et demandent automatiquement la retransmission du bloc dégradé. Il existe un de ces logiciels dans le domaine public, il s'agit de **KERMIT** qui est disponible sur les **ATARI 8 et 16 BITS.**

4 - LE CAS TRANSPAC

Il existe un cas particulier, **TRANSPAC**. La transmission entre abonnés **TRANSPAC** se fait par paquet avec une très haute fiabilité, mais il a été prévu un moyen d'accès pour les utilisateurs non raccordés à ce réseau spécialisé. Il est appelé **Point d'Accès de Données (PAD)**, le Point d'Accès Vidéotex **(PAV)** étant un **PAD** particulier. Ces dispositifs se chargent de transformer les communications asynchrones **300-300, 1200-1200 ou 1200-75** en mode caractères avec un simple contrôle de parité paire en transmissions synchrones par blocs habillés (paquets) avec des contrôles plus poussés.

Jean-Michel DUBOIS

DERNIERE HEURE...

4 livres édités par MICRO APPLICATION sortent pour le 520 ST courant octobre.

Le livre du GEM sur ATARI ST : 149 F.
Le livre du langage machine de l'ATARI ST : 149 F.
et courant novembre
ATARI Trucs et Astuces : 149 F.
La bible de l'ATARI ST : 249 F.
En plus du magazine, vous n'allez pas manquer de littérature.

ATARI FRANCE distribuera courant décembre un kit de **digitalisation couleur** au prix très étudié de **4.950 F.** Ce merveilleux joujou se branchera sur l'entrée **MODEM** et acceptera, vous l'avez bien lu, **n'importe quel** signal VIDEO.

ET PRATIQUE

« BRANCHEZ-VOUS ! » COMMENT ET OU SE CONNECTER ?

Après un petit peu de théorie, voyons voir si vous allez autant briller dans la pratique...
Nous allons enfin découvrir comment pénétrer dans le monde ésotérique des Réseaux !

La recette est en fait très simple : pour se connecter à un serveur, vous avez besoin d'un ordinateur (c'est aujourd'hui le 520 ST), d'un modem et d'un logiciel de communication. Commençons par examiner les divers modems proposés sur le marché, puis voyons un peu sur quels serveurs on va pouvoir essayer ces petites bébêtes...
Comme vous le savez, il existe en France deux normes très utilisées dans le domaine des communications : La norme **V21** et la norme **VD3** . La première correspond à une vitesse de transfert de **300bauds** en full-duplex, c'est-à-dire que l'émission et la réception se font simultanément à la même vitesse : la deuxième correspond à une vitesse de **1200 bauds** en réception et de **75** bauds en émission. Les serveurs **300 bauds** ne vont transiter que des caractères ASCII. Les serveurs en **1200/74** utilisent la norme **VIDEOTEX** utilisée par le **Minitel** .
La quasi-totalité des modems vont donc fonctionner en **V21** : les modèles plus performants (mais plus chers...) vont permettre également le **V23** . Mais quels modems peut-on connecter sur le **520 ST** ?
Le **« Jackintosh »** possédant une interface série, tout modem ayant un port **RS-232** va donc convenir. Les caractéristiques principales des modems vont entre le type de couplage (couplage direct sur la ligne téléphonique, ou bien couplage accoustique), le choix des vitesses de transfert **(300/300, 1200/1200 1200/75, etc...)**, la possibilité de composition automatique des numéros, la possibilité de réponse automatique pour faire un serveur, moyennant le logiciel adéquat, bien sur.

Voici une liste non exhaustive des principaux modems disponibles sur le marché :
— Parmi les rares modems homologués **PTT**, signalons le modem accoustique **ANDERSON-JACOBSON** en **300 bauds** (environ 3000 F...).
— Toujours en accoustique, le petit modem portable **EPSON** fonctionne très bien en **300 bauds** (environ 1400 F).
— Le modem **APPLE/SECTRAD** fonctionne en **V21** et **V23.** Possibilité des réponses automatiques en **300 bauds** (environ 2500 F chez les revendeurs Apple).
— Le **DIGITELEC 2000** + fonctionne en **V21** et **V23** pour environ 2000 F. Spécifier l'interface **SERIE** (Av. Kennedy, 33700 Merignac).
— Le **BUZZ-BOX**, pour environ 1200 F, vous permettre de vous connecter en **300 bauds.** Prévoir des piles de rechange pour les longues connexions d'hiver, sinon acheter un bloc secteur... **ULTEC**
— Signalons enfin le modem **300 bauds** le moins du marché : **790 F TTC !** Nous l'avons essayé et croyez nous, ce n'est pas le plus mauvais. Rapport qualité/prix imbattable... **DYNAMITE COMPUTER/MICRO-VIDEO**
— Pour les bricoleurs du Dimanche seulement, de très bonnes descriptions ont été faites dans les journaux d'électronique **MICRO ET ROBOTS ET ELEKTOR**, autour du fameux chip **MODEM AM 7910.**
Dans la listée des modems donnés précédemment, un seul est homologué PTT. En effet, ces modems se comptent sur les doigts de la main, et leur prix avoisine plutôt les 6000 francs que les 1000 francs ! L'utilisation d'un matériel non homologué est théoriquement interdite, mais tant que vous ne dérangez pas votre voisine dans ses conversations téléphoniques avec votre porteuse, l'administration des PTT ferme les yeux...

Vous pouvez donc choisir entre un modem pas cher, mais qui fonctionne uniquement en **300 bauds**, ou bien entre en modem plus cher mais qui accepte également le **1200/75.** Sachez cependant que le fait de posséder un modem **V23** ne vous permettre pas forcément de vous connecter sur le réseau Minitel : il vous faudra en plus un logiciel spécial d'émulation Vidéotex qui vous permettra d'afficher à l'écran les caractères semi-graphiques. (A l'heure actuelle, ce logiciel n'existe pas encore sur le **520 ST**, mais gageons qu'il le sera d'ici peu...)
Ceci nous amène bien sur à parler du logiciel de communication sans lequel rien n'est possible ! Le plus simple logiciel permet d'afficher à l'écran les caractères réçus par le port série, et d'envoyer les caractères tapés au clavier. Autour de cela viennent se greffer diverses options, comme la possibilité d'imprimer son texte simultanément, de le sauvegarder dans un buffer, de transmettre et de recevoir des fichiers, de pouvoir utiliser des **« MACROS »**, c'est-à-dire d'envoyer des séquences de caractères prédéfinies, pour effectuer une connexion automatique par exemple.
Sur le **520 ST**, il y a un logiciel d'émulation de terminal **VT 52** inclus dans le système, ce qui permet de se connecter très facilement et très rapidement. Il suffit de le selectionner en cliquant avec la souris sur **VT52 EMULATOR** dans **DESK.** En appuyant sur la touche HELP, on peut configurer le terminal en **300 bauds**, full-duplex, 8 bits de data, pas de parité ; en appuyant sur **UNDO**, on retourne au système.
Bien sûr, ce logiciel ne possède pas les fonctions décrites précédemment, mais on peut toutefois effectuer une hard-copie d'écran sur imprimante en appuyant sur la touche **CTRL HELP.**

Des softs de communication sont en train d'être développés en ce moment sur **L'ATARI :** aussi devrait-on bientôt trouver sur le **520 ST** « l'équivalent » du fameux **ASCII EXPRESS PRO SUR APPLE II,** ou bien encore de **MAC TELL 2** sur **MAC INTOSH...**

Mais que peut-on bien faire avec un modem ?

Tout d'abord, on peut communiquer avec quelqu'un possédant un autre modem. C'est donc un nouveau moyen beaucoup plus amusant et original et tellement plus fastidieux de communiquer avec ses amis...
On peut également transférer des programmes ou des fichiers. La taxation est alors la même que celle utilisée habituellement par les **PTT.**
Ensuite, on peut se connecter sur la multitude de réseaux et de serveurs qui existent en France à l'heure actuelle. (Et pourquoi pas à l'étranger, moyennant quelques impératifs techniques et surtout beaucoup de sous pour payer ses taxes téléphoniques...)
Pour l'instant, le plus important réseau est **« CALVADOS»** qui compte plus de mille abonnés. Pour vous y connecter en **300 bauds,** rien de plus simple : Composez le numéro Transpac **601 91 00,** ou bien le **16 (3) 601 91 00** si vous êtes en province.
L'appel de ce numéro ne vous coutera qu'une taxe de base, c'est à dire **75 CENTIMES** quelles que soient la durée et la situation géographique d'où vous appelez.
Une fois que vous entendez la porteuse, basculez le commutateur de votre modem sur la position **« APPEL »,** puis tapez le code d'accès **CALVADOS : 175040781,** suivi d'un **CTRL G.** Vous pouvez alors rentrer votre mot de passe confidentiel...
Les services proposés sur **CALVA** sont très complets :
— La **MESSAGERIE** créée à l'origine pour le réseau de revendeurs **APPLE** vous permet de communiquer avec n'importe quel abonné par le biais de sa boîte à lettres.
— Les **FORUMS** ou **« BOÎTES A IDEES »** sont des services de messagerie plublique où vous pouvez faire des contributions sur des sujets déterminés (programmation, jeux, communication, cinéma, humeurs, etc...).
— **LE BRIDGE...** Si vous etes amateur ou passionné de bridge, vous pourrez vous entraîner avec le Champion Michel Lebel !
— Les services **BOURSIERS** vous permettent de consulter les cotations des valeurs négociées dans le monde entier, d'avoir des informations sur les sociétés cotées en bourse. Vous pourrez même effectuer la gestion de votre portefeuille boursier !
— Les dépêches des **L'A.F.P.** vous tiendront au courant des dernières nouvelles du monde entier...
— Bien sur, vous pourrez effectuer des transferts et sauvegarder des fichiers avec gestion de vos fichiers.
— Enfin, le plus original est certainement la **« CONVIVIALITE »,** ou **« CHAT »** en anglais, où vous pourrez discuter et délirer en temps réel depuis votre terminal, avec les autres abonnés de **CALVA** connectés simultanément.
Le tout est géré par un **HARRIS 800** avec environ un gigaoctet de mémoire de masse sur disque dur...
De plus, le frontal **X25** possède une trentaine de portes sur la ligne **TRANSPAC,** ce qui permet la connexion simultanée de **30** abonnés !
L'abonnement est d'environ 200 francs par mois, ce qui n'est pas excessif compte tenu des services proposés. Cela revient peut-être même moins cher que certains serveurs **VIDEOTEX** qui vous débitent une taxe toutes les **45 SECONDES** sans que vous vous en rendiez compte.
Renseignements : **CALVADOS LIONEL LUMBROSO TEL : 705 09 04**
Parmi les petits serveurs Parisiens, en voici un qui fait pas mal parler de lui en ce moment : **FUTURA, LA VILLE DU FUTUR... TEL : 500 30 15** est accessible en **300 bauds** ou bien en **1200/75.** L'abonnement est de 150 francs pour six mois ; mais même si vous n'êtes pas abonné, vous pouvez quand même essayer en tapant le mot de passe **« DEMO ».** Le réseau fonctionne avec une carte Apple-Tell et un disque dur.
Les services proposés sont la messagerie, les forums, et surtout quelque chose d'inédit : des jeux d'arcade en Videotex !
Enfin pour finir, voici une liste (non exhaustive) de petits serveurs. Certains sont en **300 bauds,** d'autres en **1200/75,** d'autres font à la fois **V21** et **V23.** Certains ne fonctionnent que le soir, certains ne fonctionnent même pas du tout...
Libre à vous donc de les essayer pour vous « balader » un peu dans le monde des réseaux télématiques.
Bon voyage !

(Jean-François MOUSSEAU)

LISTE (PAS COMPLETE) DES MICROS SERVEURS

FUTURA	500-30-15
TELEPOM	523-55-15
95.2	538-72-49
O.U.F	531-57-25
DEVIL	738-61-44
KAEZO	962-20-01
ACTIFM	876-45-67
EVEREST	367-19-80
GUFIN	565-10-09
SESAM	806-57-59
MICROKIT	885-50-27
CHAMONIX	523-53-60
MESTEL	580-59-64
CIDER	890-23-13
M51	808-44-37
FUN SERVICE	797-67-87
MICROCOSME	758-51-56
MESTEL	874-64-88
BS	327-82-89
TROLL	663-43-96
PHAN SERVICE	797-67-87
PIC	306-66-66
SAGA	077-83-00
SUPER FLUX	581-32-92
ELLIS	204-33-74
EROTEL	874-64-88
ASCOO	241-65-00

VOUS AVEZ DIT ON-LINE ?

FUTURA

500-30-15

MINITEL/MODEM

LOGICIELS HORIZONTAUX ...

GEST

Trois programmes constituent la base de l'équipement logiciel d'un bureau :
le tableur popularise par VISICALC, MULTIPLAN ou LOTOS 1.2.3.
le gestionnaire de fichier comme PFS File.
le traitement de texte comme Wordstar.
Nous précisons brièvement ici le rôle de chacun d'entre eux.

LE TRAITEMENT DE TEXTE

Un ordinateur équipé d'unt traitement de texte présente de nombreux avantages par rapport à une machine à écrire : édition, formattage, stockage.

L'EDITION

Sur une machine à écrire, les erreurs sont définitives, seul le flacon d'encre blanche permet de les corriger avec les conséquences esthétiques que l'on connait sur l'apparence finale du document.

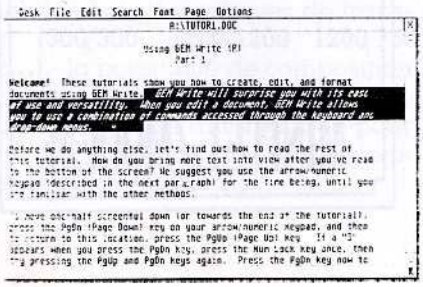
Avec un traitement de texte, le document est tapé comme il vous vient. C'est en le relisant que l'on corrige les fautes d'orthographe, les lourdeurs, les répétitions. On peut aussi supprimer des mots, des phrases, des paragraphes ou encore les changer de place.

LE FORMATAGE

Une fois qu'on est sûr de la qualité du document, on va le mettre en forme. Avec la machine à écrire on doit choisir le début de la ligne et faire attention quand on arrive vers la fin pour passer à la ligne suivante. Le document est alors fixé de manière définitive. Avec le traitement de texte, ce document peut être manipulé à l'infini, l'utilisateur fixant la marge gauche, la marge droite, la justification éventuelle, la pagination etc... Associé à une bonne imprimante on peut également demander des grosses lettres, des italiques, un soulignage, etc...

LE STOCKAGE

Dernier avantage et non des moindres pour le traitement de texte, la possibilité de sauvegarder sur disquette ou cassette les différents documents. Ceux-ci peuvent être ainsi réutilisé avec ou sans modification. On peut ainsi garder des lettres types, des adresses, des textes souvent employés.



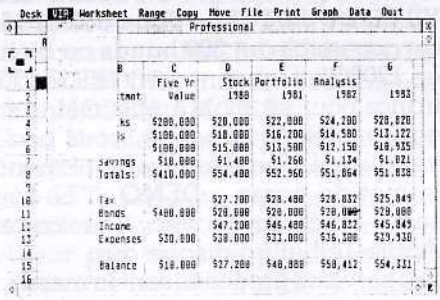
LES TABLEURS

Le traitement de texte manipule essentiellement des lettres : majuscules, minuscules, signes de ponctuations divers. Le tableur, lui, manipule des chiffres. Il se présente comme un tableau constitué de lignes et de colonnes. A chaque intersection, l'utilisateur peut entrer un titre, une valeur ou une formule mathématique.

Par exemple, une banque pourra faire un tableau avec les valeurs en francs de toutes les monnaies étrangères. Sur une ligne la valeur médiane. Sur la ligne suivante, la valeur à laquelle la banque vend cette devise. (Ces valeurs sont calculées à partir d'une formule qui tient compte de la commission de la banque). Avec un tableur, il suffit de taper dans la case dollar 8.52 à la place de 8.44 pour que soient automatiquement affichées les nouvelles valeurs d'achat et de vente.

Ceci permet de modeler de nombreuses situations financières : Comptabilité, paye, gestion financière.

Les applications d'un tableur en gestion sont innombrables et peu de programmes ont contribué autant au succès des micro-ordinateurs. Complètement inutile pour un certain nombre de personnes, les autres se demandent comment ils ont pu s'en passer jusque là.



	B	C	D	E	F	G
1		Five Yr	Stock	Portfolio	Analysis	
2	tment	Value	1980	1981	1982	1983
3						
4	ks	$200,000	$20,000	$22,000	$24,200	$28,820
5	s	$100,000	$18,000	$16,200	$14,580	$13,122
6		$100,000	$15,000	$13,500	$12,150	$10,935
7	Savings	$10,000	$1,400	$1,260	$1,134	$1,021
8	Totals:	$410,000	$54,400	$52,960	$51,864	$51,838
9						
10	Tax		$27,200	$28,480	$28,032	$25,849
11	Bonds	$400,000	$20,000	$20,000	$20,000	$20,000
12	Income		$47,200	$48,480	$46,832	$45,849
13	Expenses	$30,000	$30,000	$31,000	$36,300	$39,930
14						
15	Balance	$10,000	$27,200	$40,880	$50,412	$54,331
16						

LE GESTIONNAIRE DE FICHIER

Les fiches encombrent les tiroirs de nombreux bureaux, dans de nombreuses professions.

Sur leur support de papier cartonné, elles présentent de nombreux défauts :

1. L'encombrement

Un fichier papier peut occuper tout un tiroir. Le même fichier tiendra généralement sur une disquette. Si vous devez tenir de nombreux fichiers, le gain de place devient très important.

2. Le classement

Un fichier n'est utile que rangé. Il faut donc s'astreindre à toujours remettre chaque fiche enplace sous peine de rendre le fichier inefficace.

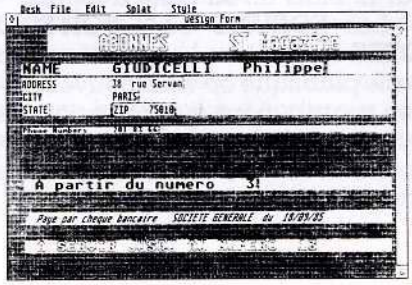
A l'inverse, un fichier informatique est toujours rangé automatiquement que vous créeiez une nouvelle fiche ou que vous en consultiez une ancienne.

3. La mise à jour

Une fiche est souvent vivante, c'est-à-dire que l'information qu'elle contient varie avec le temps. Plus vos fiches évoluent, plus elles ressemblent à de mauvaises copies d'écoliers. La fiche informatique se modifie et s'agrandit sans ratures, quelque soit le nombre de mise à jour.

4. La recherche

Mais la ou le gestionnaire de fichiers donne toute sa puissance, c'est dans la manipulation des fiches. Imaginons un fichier client avec : nom, adresse, code postal, ville, téléphone, total des sommes reçues, total des sommes dues. Avec un fichier papier, vous classerez sans doute vos clients par ordre alphabétique. Il sera alors facile de retrouver Monsieur Dupont, mais comment retrouver tous les clients du Pas-de-Calais, ou tous ceux qui vous doivent de l'argent. La seule solution est de relire toutes les fiches une par une. Avec l'ordinateur, c'est l'affaire de quelques secondes. Vous pouvez même sélectionner vos fiches sur plusieurs critères en demandant par exemple à l'ordinateur de sortir les clients qui habitent la région parisienne et qui doivent plus de 1.000 Francs. Vous pouvez bien sur avec un seul logiciel de gestion de fiches, créer autant de fichiers que vous voulez .



TION

ET VERTICAUX

Un nouveau souffle

L'apparition sur le marché de l'Atari 520 ST va permettre l'extension d'une catégorie particulière de logiciels : les logiciels verticaux.

Il existe actuellement sur le marché trois grandes catégories de logiciels qui s'échelonnent du cadre le plus général au plus particulier :

— **Les logiciels généraux :** (tableurs, traitements de texte, bases de données)

Ces logiciels ne sont liés à aucune contrainte spécifique ; bien au contraire, leurs cahiers des charges exige leur polyvalence. Ils doivent pouvoir — du moins en théorie — s'adapter à tous les problèmes. Dans la réalité, la maîtrise de ces logiciels est difficile et la réalisation d'applications demande une très longue période d'apprentissage.

— **Les logiciels verticaux :** (Gestion de cabinets médicaux, Gestion Artisants...)

Contrairement à la catégorie précédente, ces logiciels sont entièrement destinés à une profession ou à une branche d'activité.

Leur conception reste assez générale car ils doivent pouvoir satisfaire toutes les tendances au sein d'une même profession.

— **Les logiciels spécifiques :**

Cette dernière catégorie nous présente des logiciels conçus spécialement pour une personne (physique ou morale) : c'est le « sur mesure » de l'informatique. Ces logiciels sont indispensables, du moins tant que les logiciels de la première catégorie ne seront pas devenus « intelligents ».

Toutefois, ces logiciels posent des problèmes importants au niveau du temps de développement et du prix de revient puisqu'il sont vendu à l'unité.

Les logiciels verticaux sont donc un bon compromis entre les logiciels généraux et les logiciels spécifiques.

Ils ont le très net avantage d'être opérationnels tout de suite, très peu de manipulations étant nécessaires pour les utiliser.

Etant en général conçus avec le concours de plusieurs membres de la branche d'activité, ils correspondent réellement aux besoins des futurs utilisateurs, ce qui n'est malheureusement pas toujours le cas en micro-informatique.

Qui utilise les logiciels verticaux ?

Toutes les professions regroupées autour d'une même branche d'activité peuvent bénéficier de la souplesse d'un tel logiciel.

Parmi ces groupes, ont peut citer les professions libérales (médecins, architectes...), paramédicales (infirmifers, masseurs...) ainsi que les artisants.

Il est à noter que c'est souvent à l'instigation d'une chambre professionnelle ou d'un syndicat que se développe un tel logiciel, ce qui a l'avantage pour le concepteur de trouver un débouché immédiat et officiel pour son travail.

Toutes les personnes citées ont au moins un point commun : elles sont le plus souvent installées à leur compte et leur travail administratif est très lourd. Le plus souvent, elles tiennent elles-même leur compabilité pour minimiser les frais.

Toutes ces heures de travail sont

hors-activité et ne sont donc pas productives : en clair, c'est du temps perdu !
On voit donc que le besoin de traitement d'information est bien réel, ainsi en est-il de son automatisation et là, l'informatique propose sa solution sous la forme des logiciels verticaux.
On peut donc essayer de tracer un portrait du logiciel type :
— Il doit être facile à maîtriser ; le temps est un facteur important dans le type de professions visées, comme nous l'avons vu précédemment.
— Il doit tendre à supprimer le plus possible le travail redondant et ininteressant, quitte à ne pas effectuer un traitement plus subtil mais qui n'est effectué qu'une fois par mois.
— Il doit éviter les manipulations informaiques « traditionnelles » (sauvegardes compliquées des fichiers, régénérations des index... etc) pour une approche plus « douce » du contact homme-machine.
On constate donc que la principale caractéristique d'un tel logiciel est la facilité de manipulation, ce qui nous amène à la notion délicate d'interface utilisateur.
Cette interface permet le dialogue entre l'homme et la machine : plus cette interface est sophistiquée, plus le contact sera facile.
Avec l'apparition de systèmes comme l'Atari 520 ST, qui sera à la portée de toutes les professions, les logiciels verticaux vont connaître un grand développement.
En effet, les logiciels utiliseront l'interface utilisateur intégrée de cette machine, ce qui pour le « consommateur » équivaut à une standardisation des programmes (les menus sont identiques, la manipulation des fenêtres est la même pour tous les programmes... etc).
Voilà donc une solution au problème de la simplicité d'utilisation de ces programmes.
Quant au programmeur, ce sera à lui de veiller à ce que son programme respecte scrupuleusement les règles de cette interface.
Il y a donc fort à parier que de nombreux logiciels verticaux seront disponibles très rapidement sur cette machine.
En conclusion, l'utilisateur a maintenant un large éventail de choix possibles au niveau des logiciels :
— Acheter un logiciel général, le maîtriser et construire lui-même son application (solution fort instructive mais couteuse en temps).
— Se regrouper autour d'un syndicat ou d'un noyau d'une vingtaine de personnes et faire développer un logiciel vertical.
— Faire développer un logiciel spécifique : c'est de loin la solution la plus onéreuse.

NYMPHIR

NYMPHIR est un programme permettant la gestion d'un cabinet d'infirmier.
NYMPHIR gère un fichier client, chaque fiche comportant 27 critères plus les archives des soins.
Il imprime les formulaires de sécurité sociale et met à jour différents fichiers tel que les règlements comptants.
NYMPHIR imprime des états récapitulatifs des paiements des mutuelles triés par numéro de mutuelle et par ordre alphabétique.
NYMPHIR, disponible très prochainement...
Chez COGITAL
6, rue MIOLLIS
06000 NICE, Tél : (93) 87.34.95

IMPORTATEURS DISTRIBUTEURS

Nous francisons tous vos manuels et/ou vos programmes sur ST.
Contactez
PRESSIMAGE
210 rue du Faubourg St-Martin
75010 PARIS **Tél : 239.09.21**

AU BANC D'ESSAI

EXPRESS Letter Processor.
de Mirage Concepts, Inc.

NOUS AVONS AIME :
— La possibilité d'introduire des caractères spéciaux pour commander l'imprimante.
— La possibilité de récupérer les fichiers d'autres programmes.
— Le bon fonctionnement du mélange adresse/lettre avec la possibilité de sélection de n'importe quelle adresse dans le fichier.

NOUS AVONS REGRETTE :
— La lourdeur d'utilisation des paragraphes.
— L'absence de fonction SEARCH (Recherche)
— L'obligation faite de quitter le fichier sans oublier de taper une option.

Premier logiciel à avoir atteint nos bureaux, Express L.P. rassemble les fonctions d'une machine à écrire électronique et d'un traitement de texte. Un fichier d'adresses et des possibilités de communication viennent s'ajouter à ce programme très complet.

Avant de commencer, l'éditeur recommande de faire une copie de sécurité ce qui sous-entend que le programme n'est pas protégé. Encore deux manœuvres préliminaires, le formatage d'une disquette pour la sauvegarde du texte et des fiches d'adresse. Et le parmètrage du programme en fonction de l'imprimante utilisée.

LA MACHINE A ECRIRE ELECTRONIQUE :

Destinée à ceux qui veulent travailler tout de suite, cette fonction s'apprend très vite. Si vous sélectionnez cette option, vous obtenez une page blanche surmontée d'une réglette numérotée de 0 à 78.
Vous pouvez taper une ligne puis la corriger. Elle ne s'imprime que quand vous passez à la ligne suivante.
Il est possible d'envoyer des caractères de contrôle directement du clavier pour donner des instructions spéciales à l'imprimante : caractères plus larges ou plus étroits, italiques, gras, exposants,...
Il faut bien evidemment vérifier que votre imprimante possède ces modes spéciaux.
A l'aide de la réglette, vous pouvez facilement fabriquer des étiquettes, en vous repérant avec la réglette de votre imprimante.

LE TRAITEMENT DE TEXTE :

La page est formatée de façon à avoir une marge gauche à 7, une droite à 72 et 66 lignes de haut en bas. Ces caractéristiques peuvent être bien sûr changées au gré de l'utilisateur. Les paragraphes doi-

vent impérativement être séparés par une ligne blanche pour être correctement édités et formatés. Ce qui entraîne une certaine perte de temps pour ceux qui ne désire pas cette ligne. Par contre la façon de charger des textes depuis le disque est bien intéressante. Le texte vient automatiquement s'afficher en fin de tout texte préexistant. Ce qui est bien pratique pour mélanger des textes à sa convenance.
L'éditeur est assez complet, avec des possibilités de formatage partiel. La majorité des codes de contrôle s'affiche quand on presse la touche Help. Le soulignage et les caractères gras y figurent, ainsi qu'un signe d'arrêt d'impression qui met le programme en fonction machine à écrire. Ceci permet d'intervenir lors d'une impression pour ajouter une quelconque information.

LE FICHIER D'ADRESSE :

C'est un des points forts de ce logiciel, il permet l'insertion d'adresses et de formules de politesse automatiquement dans des lettres types. En effet si lors d'une impression le programme rencontre des codes spéciaux, il va vous demander quelle adresse vous voulez inclure dans votre courrier. Si vous tapez DUPONT, il vous sortira l'adresse de DUPONT. Si vous tapez DU il sortira DUPONT mais aussi DURAND, DUBOIS, etc...
En tapant 60 vous aurez toutes lesfiches de l'Oise ce qui est pratique pour des mailings groupés.
Il est ainsi possible de sélectionner une fiche sur n'importe quel champ de la fiche.
Trois fichiers sont inclus avec le programme pour des impressions spécifiques. EXLIST permet d'obtenir un listing du fichier. EXLABEL et EXENV permettent l'impression d'étiquettes et d'enveloppes. Ces fichiers s'appellent avec un simple code à partir du traitement de texte.

LE LOGICIEL DE COMMUNICATION :

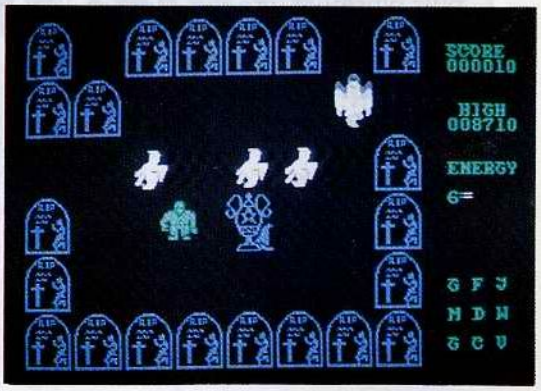
Si vous avez connecté un modem (voir nos différents articles dans le présent numéro), E.L.P. peut vous aider à communiquer. Vous pouvez envoyer ou recevoir des fichiers, une fonction intéressante étant « Disk Transfer » qui permet, quand on lit des messages, de pouvoir ne recevoir que ceux qui vous intéressent.

FONCTIONS DIVERSES :

Vous pouvez lister les fichiers, les copier, les renommer, les effacer directement à partir du programme.
Un fichier supplémentaire EXMODIFY donne une liste d'imprimantes pour lequel le programme peut automatiquement se configurer. Si vous ne posséder pas une imprimante de la liste, vous pouvez configurer le programme manuellement, en indiquant à l'ordinateur les codes qui sélectionnent telle ou telle fonction sur votre imprimante. Une petite compréhension des codes ASCII et la lecture assidue du manuel de l'imprimante sont requises.
EXPRESS L.P. n'a manifestement pas été écrit spécialement pour le 520 ST. En fait, il ne fait pas appel à l'environnement **GEM.**
Mais l'avantage est de pouvoir disposer tout de suite d'un programme qui tourne. Ceux qui travaillent avec leur ST y seront sensible, ainsi qu'à son prix qui est très raisonnable.
ATTENTION ! Si vous obtenez bien des caractères accentués sur l'écran, vous ne les obtenez pas sur l'imprimante dans la version standard de E.L.P.. Il faut absolument disposer d'un utilitaire qui redirige ces caractères dans la bonne direction sous peine d'être obligé de taper des séquences de codes étranges pour avoir des lettres accentuées.

THE LANDS OF HAVOC MICRODEAL/Import.

En attendant des programmations ayant spécialement le 520 ST comme cible, les amateurs de jeux d'aventure et de jeux d'adresse seront satisfaits de pouvoir dès maintenant nourrir leur machine avec « The Lands of Havoc », un jeu de Microdeal initialement écrit pour le QL de Sinclair.
Les premiers décors font irrésistiblement penser à un écrant QL. Nous ne les avons pas tous vus, bien sûr, car il y en a environ 2000. Mais neuf cartes sont fournies avec le jeu qui permettent d'aider les joueurs à se dépétrer des 81 premiers tableaux.
Une contrée, jadis florissante, est tombée sous l'emprise des forces des Ténèbres. Après bien des siècles d'obscurantisme, un espoir luit pour ce peuple opprimé, grâce à vous. Mi-homme, mi-bête vous allez vous battre contre une galerie d'entités aussi hideuses les unes que les autres. A ce propos, après quelques salles, le jeu devient si frénétique qu'on se demande si c'était bien là le désir du programmeur, ou s'il s'agit d'un effet secondaire dû à la rapidité du ST par rapport au QL.
Chaque contact avec les bestioles baissent votre niveau en énergétique et les premières prties s'achevènt régulièrement par le message « Vous avez complété 0,00 % de l'aventure ».
Le chemin pour détruire les Seigneurs de l'Ombre est indiqué dans un coffre. Le secret pour l'ouvrir est dans le Livre du Changement. (Ce jeu arrivant d'Angleterre, n'y voyez aucune allusion politique). Au fur et à mesure que vous avancez dans l'aventure, de nouvelles clefs sont révélées qui vous permettent de poursuivre votre quête. Ce jeu restera dans l'histoire comme le premier disponible pour le 520 ST. Et même



si on a l'occasion dans quelque temps de voir le ST exploité plus complètement, il peut permettre un break salutaire aux programmeurs en butte avec leurs premiers bugs en langage C ou à soulager des maux de tête nés des premières utilisations professionnelles.

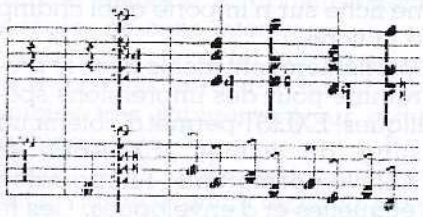
IL EST MIDI, DOCTEUR SCHMURBLUK

En plus d'interfaces classiques : série/parallèle/disque dur, les notices techniques du ST mentionnent l'interface MIDI. Pour certains professionnels, MIDI fait partie de la vie quotidienne, pour les utilisateurs de micros, il reste souvent un mot barbare au sens inconnu. Cette série d'articles aura pour but de faire découvrir l'intérêt de cette interface pour la première fois disponible en standard sur un micro-ordinateur.

Si les informaticiens ont quelques fois des problèmes pour faire communiquer les micros entre eux, les musiciens en ont tout autant avec leurs instruments. Le besoin d'une interface a occupé l'esprit de beaucoup de constructeurs depuis plusieurs années. Mais en 1981, les choses se précipitent. SEQUENTIAL CIRCUIT, la plus persévérante des sociétés américaines, propose aux autres constructeurs une entente sur la base de sa propre interface, l'USI. Les japonais furent les plus intéressés, et une synthèse avec leurs propres propositions amena en 1982 la création de la norme MIDI. Le premier synthetiseur équipé d'une interface MIDI, le Prophet 600 de SEQUENTIAL, fut bientôt disponible. Depuis, un nombre important d'instruments, du clavier à la boite à rythme en passant par les expanseurs et les séquenceurs sont munis de cette fameuse interface.

L'ordinateur vient se joindre à cette déjà nombreuse famille. Deux conséquences majeures en découlent.

Les musiciens professionnels vont pouvoir disposer, surtout avec la puissance et la rapidité du 520 ST, d'un nouvel outil dans la recherche



et la création de nouvelles musiques. Quant aux utilisateurs des micros, ils vont disposer de nouveaux périphériques qui peuvent accroître le champ de leurs loisirs informatiques.

LES NOUVEAUX INSTRUMENTS

Une imprimante est un périphérique qui ne peut que recevoir des informations. C'est le cas aussi du moniteur vidéo. A l'inverse le clavier de l'ordinateur ne peut qu'envoyer des informations, tout comme une tablette graphique, une caméra vidéo ou une manette de jeu.

Les périphériques de stockage, magnéto-cassette ou unité de disquettes peuvent indifféremment envoyer ou recevoir.

Il en est de même pour les instruments de musique électroniques. UN synthétiseur comprend un clavier et un expanseur (générateur sonore). Son utilisateur peut envoyer des informations par le clavier ou en recevoir par l'intermédiaire d'un clavier séparé, d'un autre synthe ou d'un... ordinateur. C'est pourquoi, derrière un synthétiseur équipé d'une prise MIDI il y a deux prises une « IN » qui permet au synthé de recevoir des ordres extérieurs, une « OUT » pour envoyer des informations à d'autres machines.

Un clavier nu n'aura qu'une « OUT », un expanseur qu'une « IN ».

Certains appareils sont équipés d'une MIDI « THRU » qui permet la traversée des informations MIDI vers d'autres instruments.

BRANCHER VOTRE PREMIER INSTRUMENT

Avec le ST, rien de plus simple, il ne vous faut que deux cordons DIN/DIN tout à fait ordinaires.

Toute l'interface MIDI est déjà sous le capot de l'unité centrale. Deux avantages : d'abord vous économisez les 1.000 à 2.000 francs que coûte cette interface pour les autres ordinateurs. Et ensuite, tous les utilisateurs de 520 ST ayant cette interface en standard, les éditeurs de programmes MIDI auront des espoirs de diffusion importante, ce qui va permettre l'essor de ces logiciels encore trop restreints.

LISTING APPAREILS MIDI

— Orgues Electroniques	
JVC KB-600	7.900 F
CASIO CT 6000	10.000 F
YAMAHA PS 6100	12.500 F
— Synthétiseurs	
CASIO CZ-101	5.200 F
KORG Polv 800	7.500 F
PROPHET Max	7.900 F
ROLAND Juno 106	8.700 F
YAMAHA DX7	13.500 F
SIEL Opera 6	13.900 F
— Piano Electrique	
ROLAND HP 400	11.000 F
TECHNICS PV10	9.500 F
— Guitare	
ROLAND GR700/707	?
— Accordéon	
CAVAGNOLO Midv5	6.000 F
— Boite à rythmes	
ROLAND TR 707	5.000 F
YAMAHA RX 15	5.000 F

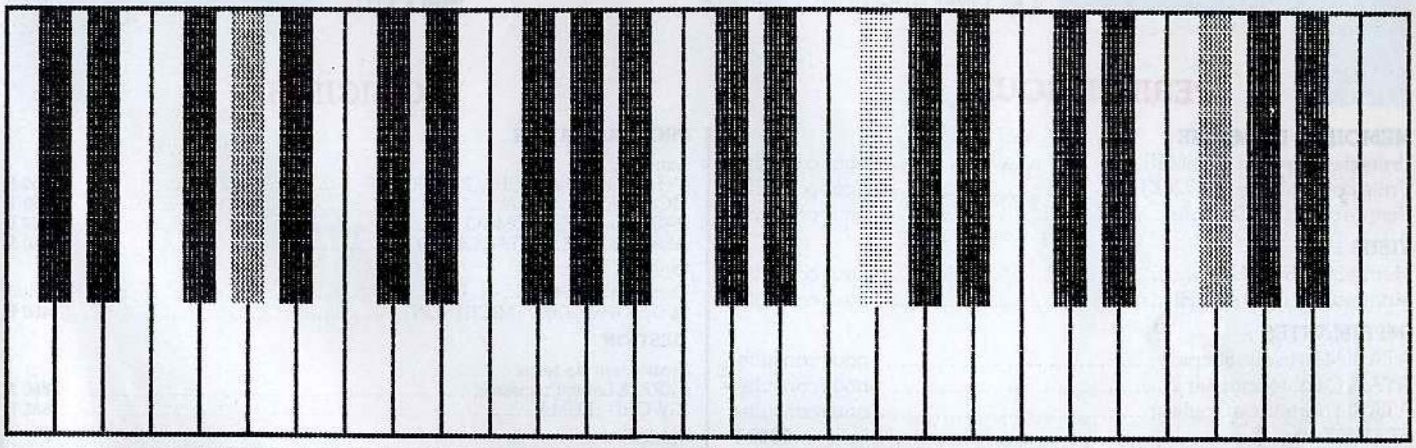
A noter que des claviers sans synthe, mais avec prise MIDI sont disponibles chez ROLAND et OBERHEIM.

L'INTERFACE

Elle répond désormais à un standard très précis régi par un document « MIDI SPECIFICATION 1.0 »
Chaque instrument aux normes MIDI est équipé d'un émetteur ou d'un récepteur (ou des deux).
L'interface opère à 31.25 Kbaud, ce qui lui permet de transmettre l'équivalent de 500 notes par secondes. Si l'on divise cette quantité par les seizes canaux disponibles, c'est encore plus de 30 notes par seconde ce qui est suffisant pour la plus part des applications.
On peut donc brancher 16 appareils différents et les faire réagir tout à tour. Chacun de ces appareils possède un numéro de canal et le musicien fera précéder par ce numéro l'information qu'il veut envoyer précisément à cet appareil. Nous verrons dans le prochain article les différentes commandes disponibles sur MIDI et les possibilités qui en découlent, mais avant de constituer un orchestre complet commençons par le début.

LES PREMIERS PROGRAMMES

Le musicien professionnel a besoin de pouvoir contrôler de nombreuses machines, de stocker des morceaux en mémoire, de trouver de nouvelles sonorites, d'écrire facilement des partitions sur papier. Des logiciels permettant ces fonctions seront parmi les premiers à apparaître. Le hobbiste micro cherche à apprendre de manière agréable la musique, à être aidé par la machine lors de ses insuffisances techniques, etc... D'autres types de logiciels vont lui être spécialement destiné, surtout si la baisse de prix sur les claviers se poursuit au même rythme. (CASIO CZ 101, le premier synthe a 5.000 francs). Mais tout ceci existe déjà (pour très cher !) sur d'autres machines. Le véritable challenge sera dans l'association d'un musicien et d'un programmeur pour utiliser le potentiel du 520 ST dans de nouvelles directions.



ST : PREMIERS FRISSONS (SUITE)

80 colonnes est un modèle du genre que ce soit sur le moniteur ou le téléviseur.
Le ST accepte soit une souris à deux boutons et un joystick, soit deux joysticks. Les paddles le laisseront sans réaction.

6 — MES IMPRESSIONS SUR LE MATERIEL

La fabrication de cet appareil est excellente, la finition soignée, sans modifications de dernière minute ce qui est signe d'une étude bien réalisée. Les cordons de raccordement des disques sont un peu courts et ne permettent pas de disposer les lecteurs derrière la machine par exemple. Les cordons d'alimentations eux sont longs et heureusement car celles-ci sont externes et dans le plus pur style ATARI, c'est-à-dire un gros bloc noir ventilé sans échauffement notable pour l'unité centrale et un par lecteur de disquette. Enfin, au prix où est l'engin il nous reste de quoi acheter une prise multiple.

7 — LES LANGAGES DE DEVELOPPEMENT

La configuration développement qui est en ma possession est livrée avec un compilateur C, l'assembleur 68000 et le linker. Avec des machines 16-32 bits comme le ST, il est probable que le C va petit à petit détrôner l'assembleur même macro dans le développement des programmes. La raison principale est la portabilité élevée des codes sources d'une machine à l'autre même avec des microprocesseurs différents. Seuls les modules d'entrée-sortié et les liens avec le système d'exploitation devant être modifiés. Ces modules de plus sont généralement livrés avec le compilateur. Le code généré par un compilateur C est un code de source assembleur qu'il est donc possible d'optimiser avant d'assembler, bien que ce ne soit nécessaire que s'il se pose des problèmes critiques de taille ou de vitesse. L'assembleur devrait se voir cantonné progressivement au développement de logiciels très pointus. Le Basic et le Logo de très belle facture. Ce sont des logiciels multifenêtre, avec un éditeur pleine page, une fenêtre pour le débuggage ou le mode trace et une fenêtre graphique. Le Logo est déjà disponible, reste à tester le Basic. Ce devrait être fait pour le prochain numéro.

Jean-Michel DUBOIS

Toute reproduction de textes et documents, même partielle, est interdite. L'envoi de textes, photos ou documents implique l'acceptation par l'auteur de leur libre publication dans le journal. Aucun document ne sera retourné. Imprimé en France. Matot Braine Reims.

LA BOUTIQUE DU ST

ARRIVAGE CHAQUE SEMAINE, CONSULTEZ-NOUS AU (1) 201 24 30, 42 01 24 30 (à partir du 27 Octobre) POUR CONNAITRE LES DERNIERES NOUVEAUTES.

Configuration 520 ST
Unité centrale 512 K
Unité disquette simple face
Moniteur Noir et Blanc (640=400)
Souris
Avec GEM, TOS (CP/M68K)
Langages BASIC et LOGO
Utilitaire graphique GEM PAINT
Traitement de texte GEM WRITE
Emulateur terminal VT-52
Recopie d'écran

LA CONFIGURATION COMPLETE ..... **8390 F HT**
LA CONFIGURATION COMPLETE ..... **9950 F TTC**

PERIPHERIQUES

MEMOIRES DE MASSE :
Unité de disquette SF (360K) ..... nous consulter
Unité de disquette DF (720K) ..... nous consulter
Disque dur 10M (Octobre) ..... nous consulter

VISUS :
Moniteur N/B ATARI ..... nous consulter
Moniteur couleur ATARI ..... nous consulter

IMPRIMANTES :
ATARI Matricielle 80cps ..... nous consulter
ATARI Qualité courrier ..... nous consulter
ATARI Thermique Couleur ..... nous consulter
STAR SG 10 ..... **3950 F**

MODEM :
(300/300) utilisable directement ..... nous consulter

CABLES :
ST/Imprimante Centronics ..... **260 F**
ST/Téléviseur Péritel ..... **295 F**
ST/Modem ..... **260 F**

HOUSSES :
Pour Unité de disquette ..... **125 F**
Pour Unité centrale ..... **125 F**

LIVRES :
LE NOUVEL ATARI ST : MICRO APPLICATION ..... **129 F**
ATARI ST COMPANION : SUNSHINE BOOKS ..... **145 F**
LE LANGAGE C : PSI ..... **100 F**
PROGRAMMATION EN ASSEMBLEUR 68000 : PSI ..... **120 F**
LE LIVRE DU LANGAGE MACHINE SUR ST : M/A ..... **149 F**
LE LIVRE DU GEM SUR ATARI : PSI ..... **105 F**

LOGICIELS :

PROGRAMMATION

Langage
4×FORTH Level 1 : DRAGON GROUP ..... **1350 F**
C COMPILER : HABA ..... **990 F**
PASCAL : METACOMCO ..... **1200 F**
ASSEMBLEUR : METACOMCO ..... **680 F**

Utilitaire
Transfert IBM PC/ST ..... nous consulter
M/DISK (Ram Disk) : MICHTRON ..... **480 F**

GESTION

Traitement de texte
EXPRES Letter Processor ..... **850 F**
K-WORD : KUMA ..... **685 F**

Tableur
VIP Professional : VIP ..... **1750 F**
K-SPREAD : KUMA ..... **685 F**

Gestion de fichier
ST BASE : MICRO APPLICATION ..... nous consulter
K-DATA : KUMA ..... **685 F**

Communication
MI/TERM : MICHTRON ..... **990 F**
K-COMM : KUMA ..... **685 F**

Gestion personnelle
HABA CHECK MINDER ..... nous consulter

GRAPHISME
DEGAS : BATTERIES INCLUDED.

JEUX
THE LANDS OF HAVOC : MICRODAL ..... **260 F**
ULTIMA 2 : SIERRA ..... nous consulter
SUNDOG THE FROZEN LEGACY : FTL ..... **490 F**
FLIGHT SIMULATOR : MICHTRON ..... **490 F**
HEX : MARK OF UNICORN ..... **490 F**
MUD PIES : MICHTRON ..... **390 F**
LES CHIFFRES ET LES LETTRES ..... nous consulter

Démonstration complète de la machine et de ses applications
le Mardi à 18 H
le Mercredi à 15 H
le Vendredi à 18 H
le Samedi à 10 H 30 et 15 H

MICRO VIDEO

8, rue de Valenciennes, 75010 PARIS - Métro : Gare du Nord

Pour commander un article par correspondance, il est recommandé de s'assurer de son prix et de sa disponibilité au préalable. (Tél : 201/24/30)
Nous acceptons Chèque, CCP et Carte Bleue. Pas de contre-remboursement. Crédit possible.